

Some parts of this thesis may have been removed for copyright restrictions.

If you have discovered material in AURA which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our [Takedown Policy](#) and [contact the service](#) immediately

GIS Linked Environmental Process Models

Thomas William Charnock

Doctor of Philosophy

Aston University

September 1997

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

Aston University
GIS Linked Environmental Process Models

Thomas William Charnock

1997

Summary

In recent years there has been a great effort to combine the technologies and techniques of GIS and process models. This project examines the issues of linking a standard current generation 2½d GIS with several existing model codes.

The focus for the project has been the Shropshire Groundwater Scheme, which is being developed to augment flow in the River Severn during drought periods by pumping water from the Shropshire Aquifer. Previous authors have demonstrated that under certain circumstances, pumping could reduce the soil moisture available for crops.

This project follows earlier work at Aston in which the effects of drawdown were delineated and quantified through the development of a software package that implemented a technique which brought together the significant spatially varying parameters. This technique is repeated here, but using a standard GIS called GRASS. The GIS proved adequate for the task and the added functionality provided by the general purpose GIS — the data capture, manipulation and visualisation facilities — were of great benefit.

The bulk of the project is concerned with examining the issues of the linkage of GIS and environmental process models. To this end a groundwater model (Modflow) and a soil moisture model (SWMS_2D) were linked to the GIS and a crop model was implemented within the GIS. A loose-linked approach was adopted and secondary and surrogate data were used wherever possible. The implications of which relate to;

- justification of a loose-linked versus a closely integrated approach;
- how, technically, to achieve the linkage;
- how to reconcile the different data models used by the GIS and the process models;
- control of the movement of data between models of environmental subsystems, to model the total system;
- the advantages and disadvantages of using a current generation GIS as a medium for linking environmental process models;
- generation of input data, including the use of geostatistics, stochastic simulation, remote sensing, regression equations and mapped data;
- issues of accuracy, uncertainty and simply providing adequate data for the complex models;
- how such a modelling system fits into an organisational framework.

These issues are explored in the thesis and recommendations are made for future projects which require the linkage of GIS and process models.

Keywords:

GIS Environmental process models Linkage Groundwater Input data

Acknowledgement

I would like to acknowledge the help of:

Kevin Voyce, a geologist at the National Rivers Authority (now the Environment Agency) for providing data and advice about the Shropshire Groundwater Scheme. Professor Kenneth Rushton of Birmingham University for advice about groundwater modelling. All contributors to the GRASS mailing lists, for numerous invaluable pieces of advice and information. Everyone who has contributed applications to the the GRASS GIS. My supervisor Dr Peter Hedges and Dr John Elgy of Aston University, Department of Civil Engineering, for help advice and support during the whole of the project. Kamarul Nasir, Jane Davenport, Chris Murphy, Robert Poole and all my other colleagues from the Department for help and advice,
and Dr Robert Ellis and Al Buchanan, for the support and the help.

Contents

1	Introduction	22
1.1	Structure of the thesis	23
2	Overview of project	24
2.1	The Shropshire Groundwater Scheme	24
2.1.1	History of the SGS	24
2.2	Previous work on the SGS at Aston University	28
2.2.1	Theoretical effects of drawdown on soil moisture	28
2.2.2	Field work	30
2.2.3	Laboratory studies	31
2.2.4	A technique for environmental impact assessment	33
2.2.5	Assessing total soil moisture removal	34
2.2.6	Drawdown response time	34
2.2.7	Assessing crop loss	35
2.3	Evaluation	36
3	GIS and process models	37
3.1	Models	37
3.2	Environmental process models	38
3.2.1	Lumped and distributed models	39
3.2.2	Practical models	40
3.3	GIS	41
3.3.1	Data-models and data-structures	42
3.3.2	GIS functionality	44
3.3.3	Cartographic modelling	45
3.3.4	Data quality, accuracy and error	47
3.3.5	The subject of GIS	47
3.4	GIS and environmental process modelling	48
3.4.1	Advantages of linking GIS and environmental process models	48

3.4.2	Approaches to linking GIS and environmental process models	51
3.4.3	Achieving linkage	52
3.5	System Design	56
3.5.1	General Strategy	56
3.5.2	The Geographic Resource Analysis Support System (GRASS)	60
3.5.3	<i>r.mapcalc</i>	61
3.6	Evaluation	62
4	Cartographic modelling for EIA of groundwater drawdown	65
4.1	Data	65
4.2	Delineating potentially vulnerable areas	67
4.3	Delineating actually vulnerable areas	68
4.4	Quantifying moisture loss	69
4.5	Quantifying yield loss	72
4.6	Discussion	74
4.6.1	Practicality	75
4.6.2	Evaluation of Hedges's (1989) technique	75
4.6.3	Outline of the next stage	77
5	Groundwater	80
5.1	Groundwater and groundwater modelling theory	80
5.2	Modelling experiences	83
5.2.1	Birmingham aquifer	83
5.2.2	Nottingham Sherwood aquifer	83
5.2.3	Previous modelling of the Shropshire aquifer	86
5.2.4	Evaluation	87
5.3	GIS and groundwater investigation	87
5.3.1	GIS and numerical groundwater modelling	88
5.4	Modflow and GRASS	89
5.4.1	Rivers package	91
5.4.2	Using Modflow	92
5.4.3	The Modflow – GRASS link	92
5.5	Characterising the Tern groundwater regime	98
5.5.1	Delineating a groundwater unit	100
5.5.2	Spatial distribution of hydraulic heads	101
5.5.3	Groundwater flows	105
5.5.4	Transmissivity	110
5.5.5	Storage	110
5.5.6	Recharge	110

5.5.7	River leakage	110
5.5.8	Confinement	111
5.5.9	Fault zones	112
5.6	Statistical investigation of the watertable	113
5.6.1	Recession	113
5.6.2	Statistical investigation into the effect of river location on the watertable surface	115
5.6.3	Statistical prediction of the watertable surface	129
5.7	Geostatistical investigation of the watertable surface	136
5.7.1	What is geostatistics?	136
5.7.2	Geostatistics of the watertable	141
5.7.3	Evaluation	146
5.8	Modelling the Tern area	150
5.8.1	The 1990 recession	150
5.8.2	The 1984 test pumping	152
5.9	Evaluation	161
5.9.1	The process of linkage	163
5.9.2	Using the GIS linked groundwater model	163
5.9.3	The Modflow GRASS link	164
5.9.4	Conclusion	164
6	Soil moisture modelling	166
6.1	Soil and soil moisture theory	166
6.1.1	Soil and soil mapping	166
6.1.2	Soil moisture and soil moisture modelling	170
6.1.3	Simple soil moisture models	174
6.1.4	Soil moisture flow models	176
6.2	GIS and soil	179
6.3	Accuracy, sensitivity and uncertainty of soil moisture models	182
6.4	SWMS_2D	182
6.4.1	Practical features of SWMS_2D	182
6.4.2	Characterisation of soil properties in SWMS_2D	183
6.5	The link between SWMS_2D and GRASS	186
6.5.1	Basic link	186
6.5.2	The controlling process	188
6.5.3	A graphical controlling process	189
6.6	Modelling with SWMS_2D	189
6.7	Discussion	195
6.7.1	Modelling with lower quality data	195

6.7.2	Generating maps of stochastic variables	197
6.7.3	The use of regression equations	201
6.7.4	Calibration	201
6.8	Recommendations	204
7	Crop modelling	205
7.1	Basics of plant growth	205
7.2	Crop modelling	206
7.2.1	Potential and actual evapotranspiration	207
7.2.2	Simple crop models	210
7.2.3	Regression models	211
7.2.4	CROPR	211
7.2.5	van Keulen's (1986) crop model	214
7.2.6	Crop specific models	216
7.3	GIS and crop models	216
7.4	Remote sensing of the distribution of crops	219
7.5	Implementation of a crop model in GRASS	223
7.6	Evaluation	225
7.6.1	Plants as <i>model-able</i> entities	225
7.6.2	Conclusions	226
8	Discussion	227
8.1	Introduction	227
8.2	Evaluation of the overall strategy	228
8.3	System design of the next prototype	230
8.3.1	User interface	230
8.3.2	Reconciling differing data models	231
8.3.3	The next prototype for linking a process model and GIS	236
8.4	A modelling system in an organisation	238
8.5	Data	239
8.5.1	Parameterisation	241
8.5.2	Surrogate data, error and accuracy	241
8.6	Evaluation	242
9	Conclusions	244
9.1	Specific recommendations and conclusions	245
9.1.1	Choosing and using environmental process models	245
9.1.2	Linking process models and GIS	245
9.1.3	Data	247

9.1.4 Conclusion	247
References	248
Appendices	263
A The code	263
A.1 Writing programs within GRASS	263
A.1.1 Writing GRASS C programs	263
A.1.2 Writing GRASS shell scripts	267
A.2 The <i>r.crop</i> program	271
A.3 Adapting <i>r.mapcalc</i>	279
A.3.1 Adding new operations	279
A.3.2 <i>look-up</i> functions	280
A.3.3 Normal random number functions	290
A.3.4 Simpson's rule function	292
A.4 GRASS and PERL	294
A.4.1 The <i>grass.pl</i> library	294
A.4.2 The <i>s.rms</i> program	295
A.4.3 The <i>r.swms.region</i> program	297
A.5 RDB	300
A.5.1 Using RDB	300
A.5.2 <i>drawdown</i> , <i>julian</i> and <i>time</i> scripts	301
A.5.3 Watertable extraction scripts	302
B Modflow-GRASS link documentation	305
B.1 <i>mod.edit</i>	305
B.1.1 Basic option	306
B.1.2 Block centred flow option	309
B.1.3 Well option	312
B.1.4 River option	312
B.1.5 Evapotranspiration option	316
B.1.6 Recharge option	316
B.1.7 Successive implicit procedure option	316
B.1.8 Parameters option	316
B.2 Other Modflow-GRASS link programs	320
B.2.1 <i>mod.cre</i>	320
B.2.2 <i>mod.write</i>	320
B.2.3 <i>mod.run</i>	321

B.2.4	<i>mod.xtrct</i>	321
B.2.5	<i>mod.del</i> , <i>mod.rename</i> and <i>mod.copy</i>	322
B.2.6	<i>mod.region</i>	322
B.2.7	<i>mod.d.grid</i>	322
B.2.8	<i>mod.rem</i>	323
B.2.9	<i>mod.ts</i>	323
C	SWMS_2D - GRASS link documentation	324
C.1	<i>r.swms</i>	324
C.1.1	The edit mode of <i>r.swms</i>	326
C.2	<i>r.swms.region</i>	328

List of Figures

2.1	SGS location map	25
2.2	SGS plan	25
2.3	SGS history	27
2.4	Soil moisture theory after Hedges and Walley (1983)	28
2.5	Movement of the soil moisture profile with drawdown	29
2.6	Idealised soil moisture profile	32
3.1	A simple hydrological conceptual model.	39
3.2	GIS: a system for managing and integrating different models of the world	42
3.3	GIS: abstraction from reality	43
3.4	The raster and vector data-models.	44
3.5	Overlay operation to delineate erosion	46
3.6	Operations in cartographic modelling	46
3.7	Sample input files for the SWMS.2D model	49
3.8	Data handling approaches for existing numerical models	50
3.9	Data flow diagram showing a logical way of linking models and GIS	58
3.10	Structure diagram showing hierarchy of control of GIS linked models.	59
3.11	A possible configuration of the individual component models	59
3.12	Elements of the modelling system.	64
4.1	Agricultural maps on the BBC and the SPARCstation	66
4.2	Hedges's (1989) region of analysis	66
4.3	Potentially vulnerable areas	68
4.4	Actual vulnerable areas	69
4.5	Situations when quantifying soil moisture loss	71
4.6	Calculated soil moisture loss	73
4.7	Calculated cereal yield loss	78
4.8	Effects of varying root depth parameter on calculated yield loss	79
5.1	Typical features of a groundwater system.	81

5.2	Rushton and Bishop's (1993) model of aquifer river interaction.	85
5.3	SRA (1974) groundwater model boundaries.	86
5.4	Conceptual model of the river bed in Modflow.	91
5.5	The difference between Modflow and GRASS grid structures.	93
5.6	The link between Modflow and GRASS.	94
5.7	The Modflow-GRASS link <i>block centred flow</i> module interface.	95
5.8	Methods of resampling between irregular grids and rasters.	96
5.9	Steps in converting a GRASS vector structure to a Modflow river file.	97
5.10	The river reach overlaid on a Modflow grid-square.	98
5.11	Steps in converting a GRASS site file to a Modflow well file.	99
5.12	The location and nature of boundaries in the groundwater model.	100
5.13	Watertable monitoring network.	102
5.14	Profiles of the March 1990 watertable generated with different interpolators	104
5.15	Interpolated watertable for March 1990.	104
5.16	Interpolated artesian areas	106
5.17	The River Tern near Stoke on Tern	107
5.18	Interpolated watertable for March 1990 including fixed-head boundaries.	108
5.19	Interpolated artesian areas, accounting for fixed-head boundaries.	108
5.20	Interpolated watertable for March 1990, accounting for the Tern river.	109
5.21	Interpolated artesian areas 1990, accounting for Tern river.	109
5.22	Profiles of the March 1990 watertable, generated with differing assumptions	116
5.23	Major flow directions of the groundwater system.	117
5.24	Transmissivities from 1973 and 1984 pump testing.	118
5.25	The effect of partial penetration of wells.	119
5.26	Confined storage coefficients from 1973 and 1984 pump testing.	120
5.27	Drift in the Tern area	121
5.28	Faults of the Tern area	122
5.29	The effects of faulting on the Lodgebank pump test, 1984.	123
5.30	Typical hydrographs at two monitoring wells 1989-95.	124
5.31	Histograms and normality plots of watertable data, March 1990.	125
5.32	Histograms and normality plots of watertable data, September 1990.	126
5.33	Mean watertable heights and depths in 1990.	127
5.34	Buffer zones around the river network.	127
5.35	Variation in mean watertable height with distance from river in 1990.	128
5.36	Different hypotheses of the annual variation of the watertable.	128
5.37	The LPOS raster.	130
5.38	Graphs of explanatory variables plotted against watertable height.	133
5.39	Graphs of explanatory variables plotted against watertable depth.	134

5.40	Variation of watertable with land surface height.	135
5.41	Examples of h-scatterplots.	137
5.42	Variogram surfaces for watertable height and depth.	143
5.43	Larger scale variogram surfaces for watertable height and depth.	143
5.44	Semivariogram plots for watertable height and depth	144
5.45	Large scale semivariogram plots for watertable height and depth.	145
5.46	A simple geostatistical estimation situation	145
5.47	Geostatistical modelling region and subregion.	146
5.48	Cross-semivariogram surfaces.	147
5.49	Cross-semivariogram plots.	148
5.50	Models fitted to the semivariogram plots of watertable height.	149
5.51	Profile of the watertable showing the 1990 recession	151
5.52	MORECS SMD data for 1990.	152
5.53	Comparison between the simple model and the interpolated surface.	153
5.54	Cross-sections of modelled and interpolated watertable surfaces.	154
5.55	November 1990 watertable surface modelled with a partial boundary	155
5.56	The groundwater model results after faults have been introduced	156
5.57	Results of modelling the Lodgebank boreholes	158
5.58	Observed and simple model simulation hydrographs of the Lodgebank double pump test	159
5.59	Observed and model simulation hydrographs of the Lodgebank double pump test .	159
5.60	Hodnet abstraction borehole and surrounding observation sites.	160
5.61	Cartographic modelling used to flatten the hydraulic head.	161
5.62	A script for performing a stepwise search of a parameter space to minimise the reported RMS error.	162
5.63	The parameter space formed by transmissivity and storage.	162
5.64	The results of the Hodnet pump test model	165
6.1	A typical soil profile	168
6.2	Classification of soils on the basis of particle size distribution	169
6.3	Availability of soil data in the Tern area.	169
6.4	Modes of soil water.	171
6.5	Interaction of soil particles and water	172
6.6	Stages in a drying soil.	174
6.7	Typical moisture–tension and moisture–conductivity curves.	175
6.8	Model of soil properties.	180
6.9	Feddes, Kowalik and Zaradny’s (1978) model of the root sink term.	180
6.10	Modified van Genuchten’s (1980) soil properties	184
6.11	Modified form of Feddes et al.’s (1978) model of the root sink term	186

6.12	Basic link between SWMS.2D and GRASS	187
6.13	A Tcl interface to the SWMS.2D link	189
6.14	The location of the Heath House site	190
6.15	Standards of particle size classes	191
6.16	Plan of Heath House site	193
6.17	Development of soil moisture in the root zone.	194
6.18	Observed and modelled soil moisture profiles.	196
6.19	A stochastic realisation of soil properties	202
7.1	Typical “S” shaped cumulative crop production curve	211
7.2	Typical variation of growth rate through a season	212
7.3	Asymptotes of growth rate.	213
7.4	The CROPR model.	214
7.5	van Keulen and van Laar’s (1986) crop model.	215
7.6	Generation of potential evapotranspiration.	218
7.7	Elements of remote sensing.	219
7.8	Clustering of pixels in multi-dimensional band space.	220
7.9	The SMAP classification.	221
7.10	Agricultural survey.	222
8.1	Different representation of space and time in GIS and Models.	232
8.2	Handling the vertical dimension in GIS.	233
8.3	Handling the temporal dimension.	234
8.4	The vertical dimension	236
8.5	Modified system flow diagram	237
8.6	The way objects in the dataset would be logically linked together under the second prototype	238
8.7	Life cycle of a loosely linked modelling system.	240
A.1	Screen shot showing the operation of the <i>r.section</i> program.	281
A.2	<i>look-up</i> functions in <i>mapcalc</i>	281
A.3	Passing a shell environment variable to an <i>r.mapcalc</i> function	282
A.4	RDB database structure.	301
B.1	Region definition screen.	306
B.2	Grid specification screen 1	307
B.3	Grid specification screen 2	307
B.4	Grid specification screen 3	308
B.5	Main option selection screen.	308
B.6	Element code specification screen	309

B.7	Element head specification screen	310
B.8	Stress period specification screen	310
B.9	Steady state check screen	311
B.10	Storage code specification screen.	311
B.11	Layer properties specification screen.	312
B.12	Well site file selection screen	313
B.13	Well check screen	313
B.14	Well flow specification screen.	314
B.15	Well layer specification screen.	314
B.16	River vector selection screen.	315
B.17	River reach specification screen.	315
B.18	Evapotranspiration check screen.	316
B.19	Evapotranspiration specification screen.	317
B.20	Recharge code check screen.	317
B.21	Recharge specification screen.	318
B.22	Iteration specification screen.	318
B.23	Parameter change check screen.	319
B.24	Region change screen.	319
C.1	The <i>r.swms</i> edit menu.	327
C.2	Screen shot of the <i>r.swms</i> edit interface.	327

List of Tables

4.1	Crop rooting depths	69
5.1	Univariate data statistics for watertable data.	114
5.2	Results of t-test on raw watertable height data from 1990.	115
5.3	Results of Wilcoxon test on raw watertable height data from 1990.	116
5.4	Results of the t-test on watertable height data from 1990 with outliers removed.	116
5.5	Results of the Wilcoxon test on watertable height data from 1990, with outliers removed.	119
5.6	Univariate statistics of watertable heights in January 1990 grouped by distance from river.	124
5.7	Correlation between watertable and depth and explanatory variables.	131
5.8	Timetable of Lodgebank pump tests.	154
6.1	Typical soil moisture properties.	173
6.2	Soil moisture head and tension	173
6.3	Heath House, tube 5 particle size distributions.	192
6.4	Calculated value of α	202
7.1	Statistics of elevation, distance and slope grouped by land cover	224

List of Symbols

α	Coefficient in retention curve, Equation 6.5 (L^{-1}).
α_T	Photosynthesis efficiency–temperature function.
a	An empirical parameter in Equation 2.6 (dimensionless).
a	Constant in Equation 6.22.
a_i	Weighting of observation u_i .
A	Cross-sectional area (L^2).
A	Slope of growth rate against water factor ($kg.ha^{-1}.mm^{-1}$).
AE	Actual evapotranspiration ($gm^{-2}s^{-1}$).
AE_s	Seasonal actual evapotranspiration.
AW	Available soil water.
β_0	The slope of $\frac{Y}{Y_M}$ against $\frac{AE}{PE_s}$ curve, Equation 7.8.
β_h	Factor to partition photosynthesis between harvested parts and unharvested parts (dimensionless).
b	An empirical parameter in Equation 2.6 (dimensionless).
b	Constant in Equation 6.22.
b	Ratio of $\frac{SWC}{AW}$ below which transpiration is limited.
b_j	Weighting of observation v_j .
BD	Bulk density of soil (ML^{-3}).
c_p	Specific heat of air.
C	Proportion of soil that is clay (%).
C	An $n + 1 \times n + 1$ covariance matrix between the observed variables; $V(x_1), \dots, V(x_n)$ and $V(x_1), \dots, V(x_n)$.
C_h	Differential moisture capacity $\frac{d\theta}{dh}$ (L^{-1}).
\tilde{C}_{ij}	Covariance between i 'th and j 'th random variables.
$CC_{i-\frac{1}{2},j,k}$	Conductance in j th column and k th layer between rows i and $i - 1$ in Modflow (LT^{-1}).
$CC_{i+\frac{1}{2},j,k}$	Conductance in j th column and k th layer between rows i and $i + 1$ in Modflow (LT^{-1}).

$CR_{i,j-\frac{1}{2},k}$	Conductance in i th row and k th layer between columns j and $j - 1$ in Modflow (LT^{-1}).
$CR_{i,j+\frac{1}{2},k}$	Conductance in i th row and k th layer between columns j and $j + 1$ in Modflow (LT^{-1}).
C_{riv}	Conductance of river bed (L^3T^{-1}).
$CV_{i,j-\frac{1}{2},k}$	Conductance in i th row and j th column between layers k and $k - 1$ in Modflow (LT^{-1}).
$CV_{i,j+\frac{1}{2},k}$	Conductance in i th row and j th column between layers k and $k + 1$ in Modflow (LT^{-1}).
Δ_e	Vapour pressure gradient (mbar).
$\Delta h_{i,j,k}$	Change in head in Modflow cell i, j, k in a timestep (L).
Δt	Length of Modflow timestep (T).
$\Delta r_j \Delta c_i \Delta v_k$	Width of the j th row, the i th column and the height of the k th layer in Modflow model (L).
$\Delta Y D_c$	Yield deficit of crop c (ML^{-2}).
$\Delta Y D_d$	Loss of milk yield (L).
d_i	Distance between centre of i 'th surrounding pixel and central pixel (L).
d_r	Root depth (L).
D	Depth to watertable (L).
D	$(n + 1) \times 1$ covariance matrix between $\hat{V}(x_0)$ and $V(x_1), \dots, V(x_n)$.
Dep_{mar}	Depth of watertable surface in March 1990 (L).
Dep_{sep}	Depth of watertable surface in September 1990 (L).
DEM	Land surface elevation (L).
$Dist_s$	Distance from river variable in spatial domain s .
$DIST$	Distance from river (L).
D_{ML}	Duration of moisture loss experienced by crop (T).
DRT_m	Time for the proportion m of the drainable water to be drained from the soil profile (T).
η	Random number of mean 0 and standard deviation 1.
ϵ_s	Random component representing fluctuations about systematic component in spatial domain s .
e_a	Saturated vapour pressure at actual temperature.
e_d	Vapour pressure of the air.
e_s	Vapour pressure at a height ($MT^{-2}L^{-1}$).
e_s	Saturated vapour pressure ($MT^{-2}L^{-1}$).
E	Evaporation rate (LT^{-1}).
E_0	Elevation of central pixel (L).

$Elev_s$	Land surface elevation variable in spatial domain s (L).
E_i	Elevation of i 'th surrounding pixel (L).
E_{pl}^i	Plant transpiration on day i ($mm.day^{-1}$).
Et	Evapotranspiration rate (LT^{-1}).
f	Spatially varying soil property.
\bar{f}	Mean of a spatially varying soil property.
$\gamma(\mathbf{h})$	Semivariogram measure of correlation, over a lag of \mathbf{h} .
h	Height above watertable (L).
h	Groundwater hydraulic head(L).
h	Soil moisture pressure head (L).
h	Lag distance separating paired data points (L).
\mathbf{h}	Lag distance separating paired data points (L).
h_b	Tension at air entry point (L).
h_c	Height of the capillary fringe (L).
$h_{i,j,k}$	Hydraulic head in cell i, j, k in Modflow (L).
h_k	Pressure head which produces the conductivity K_k (L).
h_o	Critical height (L).
h_s	Pressure head at air entry point, $h_s \leq 0$, (L).
H_{riv}	Stage or head of the river reach in Modflow (L).
I	Infiltration rate (LT^{-1}).
K	Hydraulic conductivity (LT^{-1}).
K_h	Hydraulic conductivity at pressure head h (LT^{-1}).
K_{rh}	Relative hydraulic conductivity at pressure head h (dimensionless).
K_{riv}	Conductivity of river bed (LT^{-1}).
K_s	The hydraulic conductivity under saturated conditions (LT^{-1}).
K_x	Hydraulic conductivity in direction x (LT^{-1}).
λ	Weighting factor for growth stage i .
λ_p	Latent heat of water vaporisation.
L	Length of river reach (L).
Λ	Fraction of day overcast.
$LPOS$	Landscape position index (dimensionless).
μ	Lagrange parameter, in Equation 5.20.
MD	Moisture debt (LT).
ML	Total moisture loss from root zone (L).
n	Exponent in retention curve, Equation 6.5 (dimensionless).

n	Number of surrounding pixels.
n	Number of data point pairs.
ϕ	Matric potential, sometimes called the <i>soil moisture pressure head</i> , caused by interaction between the moisture and the soil particles (L).
ϕ_g	The potential a point has due to its vertical displacement with reference to a datum point (L).
ϕ_{gas}	Potential arising from variation in gas pressure (L).
ϕ_{osm}	Potential arising from osmotic forces acting on the moisture (L).
ϕ_r	Respiration factor.
Φ	Soil water potential at a point (L).
P_c	Gross growth rate on an cloudless day ($kg.ha^{-1}day^{-1}$).
$P_{i,j,k}$	Net conductance of external sources and sinks that are dependant on the head in cell i, j, k in Modflow (L^3T^{-1}).
P_o	Gross growth rate on an overcast day ($kg.ha^{-1}day^{-1}$).
PE_s	Seasonal potential evapotranspiration.
q	Pore size distribution index (dimensionless).
\dot{q}	Approximate growth rate ($kg.ha^{-1}.day^{-1}$).
\dot{q}_{act}^i	Actual growth rate on day i ($kg.ha^{-1}.day^{-1}$).
\dot{q}_{pot}	Potential growth rate ($kg.ha^{-1}.day^{-1}$).
Q	Water flow rate (L^3T^{-1}).
Q	Cumulative crop yield (ML^{-2})
Q_{act}	Actual season's growth ($kg.ha^{-1}$).
Q_c	Sensible heat transfer to air.
Q_g	The change in stored energy of a water body.
$Q_{i,j,k}$	Net flow of external sources and sinks that are not dependant on the head in cell i, j, k in Modflow (L^3T^{-1}).
Q_l	Emitted long wave radiation.
Q_r	Flow from aquifer to river (L^3T^{-1}).
Q_{riv}	Net flow in Modflow from river reach to cell (L^3T^{-1}).
Q_{rs}	Reflected short range radiation.
Q_s	Incoming short range radiation.
Q_v	Energy transfer between water and bed.
Q_E	Energy available to evaporate.
ρ	Density of air.
r	Albedo, the proportion of incoming radiation that is reflected, albedo for short cut grass is usually taken as 0.25.

r_a	Net resistance to diffusion through air (sm^{-1}).
r_s	Net resistance to diffusion through leaves and soil surface (sm^{-1}).
R	River coefficient in Equation 5.5. (L^2T^{-1}).
R_i	Incoming radiant energy.
R_n	Net radiation (Wm^{-2}).
R_o	Out going radiant energy.
s	Spatial domain of a variable.
s	Standard deviation of a spatially varying soil property.
S	Proportion of soil that is silt (%).
S	Storage coefficient (dimensionless).
S_c	Soil coverage by the crop.
Sd	Percentage of soil that is sand (%).
Sd_c	Percentage of soil that is coarse sand (%).
Sd_f	Percentage of soil that is fine sand (%).
Sd_{vc}	Percentage of soil that is very coarse sand.
S_e	Relative soil moisture content (dimensionless).
S_{ek}	Relative soil moisture content using θ_k (dimensionless).
Sf	Surface storage of moisture (L).
$Sfth$	Surface storage threshold (L).
S_h	Sink term to account for root loss (LT^{-1}).
Sl	Soil storage of moisture (L).
$Slope_s$	Slope variable in spatial domain s .
$Slth$	Soil storage threshold (L).
S_o	Critical tension (L).
S_s	Specific storage, the volume of water released from unit volume of saturated aquifer by unit reduction in hydraulic head (L^{-1}).
$Ss_{i,j,k}$	Specific storage in Modflow model cell i, j, k . (L^{-1}).
SWC	Soil water content (dimensionless).
θ	Soil moisture content by volume (%).
θ	Set of fixed parameters estimated from data in Equation 5.9.
θ_a	Moisture content such that $\theta_a \leq \theta_r$ in Equation 6.18 (dimensionless) .
θ_h	Moisture content at pressure head h (dimensionless).
θ_m	Soil moisture content such that $\theta_m \geq \theta_s$ in Equation 6.18 (dimensionless).
θ_s	Saturated soil moisture content (%).
θ_r	Residual soil moisture content (%).
t	Moisture tension at a point in the soil (L).

t	Time (T).
T	Transmissivity (L^2T^{-1}).
T_i	Transpiration for growth stage i .
T_{Mi}	Maximum transpiration for growth stage i .
u	Wind speed at a height (LT^{-1}).
\hat{u}_0	Estimate of variable at location 0.
$\frac{dh}{dx}$	Instantaneous hydraulic gradient.
u_i	Observation of variable at location i .
u_i	Second point in i 'th data pair, separated from v_i by vector \mathbf{h} .
v_i	First point in i 'th data pair, separated from u_i by vector $-\mathbf{h}$.
v_j	Observation of secondary variable at location j .
v_T	Standardised spatial soil property.
$V(x)$	The true random variable at location x .
$\hat{V}(x)$	The estimate of the random variable at location x .
w	Saturated thickness of the aquifer (L).
\mathbf{w}	An $(n + 1) \times 1$ matrix of weights.
w_i	Weighting of the i 'th random variable.
W	Width of river (L).
Wt_{mar}	Watertable height in March 1990 (L).
Wt_s	Watertable depth variable in spatial domain s (L).
Wt_{sep}	Watertable height in September 1990 (L).
ξ	Empirical growth rate constant.
x	x co-ordinate (L).
x_i	Location of i 'th random variable.
y	y co-ordinate (L).
Y	Dry matter yield ML^{-2} .
Y	Maximum dry matter yield ML^{-2} .
z	Vertical co-ordinate (L).
z_b	River bed elevation (L).
z_r	River surface elevation (L).

Chapter 1

Introduction

The original aim of this project was to use environmental process models to assess the effect of groundwater abstraction on soil moisture and on crop development. As such it has been a process of synthesis — one that brings to bear tools, techniques and ideas from many different disciplines on to this single problem. To achieve this effectively, a medium within which all these diverse components can be brought together was required - the medium chosen was a Geographic Information System (GIS). Since beginning the project the emphasis has changed to an evaluation of the underlying philosophy of the integration of models with GIS and the issues that this raises.

The project takes as a focus the Shropshire Groundwater Scheme, which is described in detail in Chapter 2. Briefly the Shropshire Groundwater Scheme is an ongoing development of the groundwater potential of the Shropshire aquifer for augmenting flow in the Severn during drought. A problem identified by Walley (1979), prior to the public local inquiry, was the possible effect the drawdown might have on soil moisture conditions where the watertable is close to the surface.

Hedges (1989) performed an analysis in an attempt to quantify the spatial distribution and economic impact of such effects. He suggested that his analysis, which was done before they were generally available, could be performed within a GIS. He also suggested that linking together relevant numerical process models of the groundwater, soil moisture and crop systems might provide a better way to assess the effects of what is a dynamic system, rather than statically comparing two points in time as he did.

The research has developed to encompass a broader context than that discussed above. The integration of GIS and environmental process modelling is a current area of effort within the GIS community, and this project takes the approach of linking several off-the-shelf models together following a strategy that is principally constrained by an assumption of very limited resources, both money and time.

The theme of resources and cost will be taken up in later chapters, however it is important to note that when undertaken a modelling exercise for whatever purpose economic considerations cannot be considered as secondary. If it is assumed that the whole modelling effort is directed, as

it frequently is, to making a better decision and the value of the decision is less than the expense of modelling then the decision to invest resources in the modelling exercise is not valid. Where a decision is to be made regularly, as in some on going large operation, then considerable investment in a modelling system is justified, but when a decision is taken once only the expense may not be justified. This idea is expanded in Section 3.4.2. However for the moment it should be clear that although this is an academic project and so resources, both in time and money, are limited, this is not a drawback as it adds a dimension of practical realism to the conclusions that are drawn.

1.1 Structure of the thesis

Chapter 2 gives an overview of the project and describes: the Shropshire Groundwater Scheme which is the focus; the work of Hedges (1989) and the recommendations that he made. Chapter 3 discusses the many shapes and forms that models and modelling take in this project and how the technology of GIS (itself a model) can act to mediate between many of them. Chapter 4 repeats the essential features of the analysis of Hedges but using a standard GIS rather than specially written code. The experience from Chapter 4 acts as a starting point for investigating and modelling the groundwater, soil moisture and crop systems — Chapters 5, 6 and 7 respectively. Though the project did not proceed exactly as originally envisaged some very useful discussion points, recommendations and conclusions can be drawn — Chapter 8.

Chapter 2

Overview of project

This chapter describes the Shropshire Groundwater Scheme (SGS), the past involvement of researchers from Aston University with the SGS, and the reasons behind this current project.

2.1 The Shropshire Groundwater Scheme

The Shropshire Groundwater Scheme (SGS) is the focus for this project. It is a groundwater development in the UK county of Shropshire, Figures 2.1 and 2.2, designed to augment flow in the River Severn during periods of drought with water pumped from the shallow sandstone aquifer of the region. Of particular interest is the River Tern basin within the SGS which has been the subject of several previous studies within Aston University, (Hedges and Walley, 1983; Hedges, 1989). Here the conditions are such that it is possible that a lowering of the watertable will decrease the soil moisture available to crops.

2.1.1 History of the SGS

The SGS began in the early nineteen seventies when it was realised that the Llyn Clewedog reservoir would not be able to guarantee minimum flows in the Severn beyond a projected date of 1978. The Severn River Authority (SRA) proposed that a major new regulating reservoir be commissioned in the Dulas valley in Wales. There was a public inquiry in November 1971, following which the Secretary of State for Wales ruled that there be no further investigations.

Consequently the SRA put forward proposals to investigate the use of Shropshire groundwater to regulate the River Severn. The proposed investigation was to take the form of two pilot developments, one in the River Tern valley and one in the River Perry valley. At a public inquiry in 1971 the Tern pilot project was approved, the Perry project was to proceed later and in the light of lessons learnt from the Tern pilot project.

The Tern investigation began in 1971 and the Perry investigation in 1974. In both areas the



Figure 2.1: The location of the Shropshire Groundwater Scheme (SGS). Grid given in 100s of kilometres.

Figure 2.2: The plan of the Shropshire Groundwater Scheme (SGS), showing the current state of development. Grid given in 10s of kilometres.

hydrogeology, surface hydrology, soil moisture, aquatic ecology and agriculture were examined. The hydrogeological investigations included drilling and test pumping several boreholes, and monitoring the effect on local and regional watertable levels and on soil moisture.

In 1977 the Severn Trent Water Authority (which had assumed responsibility for the Severn from the SRA) put forward proposals for a fully operational groundwater development (STWA, 1977). In summary the proposals were:

- to develop 58 new and 10 existing boreholes, in eight borehole groups;
- to stage the development of borehole groups in a flexible way so that supply just met demand;
- to construct 38 new observation boreholes and maintain 53 existing ones;
- the construction of pipelines, ancillary works, control buildings and a telemetry network.

In September 1979 there was a public inquiry in Shrewsbury, Shropshire, to decide whether to recommend an operational system to the Secretary of State (Gray, 1980). The development was approved in 1981 (Musgrave, 1981) but with several changes;

- groundwater levels and soil moisture in the Tern area should be comprehensively monitored;
- development of later stages in the Tern area should only proceed if “justified by the monitoring results” (Musgrave, 1981);
- the order of the development of borehole groups was changed.

These changes were largely as a result of objections raised by Mr. Walley (Walley, 1979) on behalf of concerned farmers, see Section 2.2.

In 1989, with the privatisation of the water industry, responsibility for the scheme passed to the National Rivers Authority (NRA), and in 1996 the NRA was absorbed into the Environment Agency.

Figure 2.2 shows the current status of the SGS development. Since the inquiry, projected water demand has decreased and the commissioning of stages has slowed. Tern Stage 1 was commissioned in 1984 and the Perry Montford group of boreholes was commissioned in 1992. Work is continuing on the Leaton Stage and this is scheduled to be available in 1998, all other stages are not expected to be needed before 2000 (NRA, 1994).

In the history of the SGS there have been three drought events that have required the operation of the scheme for river regulation, these were in 1984, 1989 and 1995. Figure 2.3 shows the major events in the history of the scheme. For more information on the history see Hedges (1989), SRA (1971), SRA (1974), STWA (1975), STWA (1976), STWA (1977), STWA (1978), NRA (1985), NRA (1989) and NRA (1994)

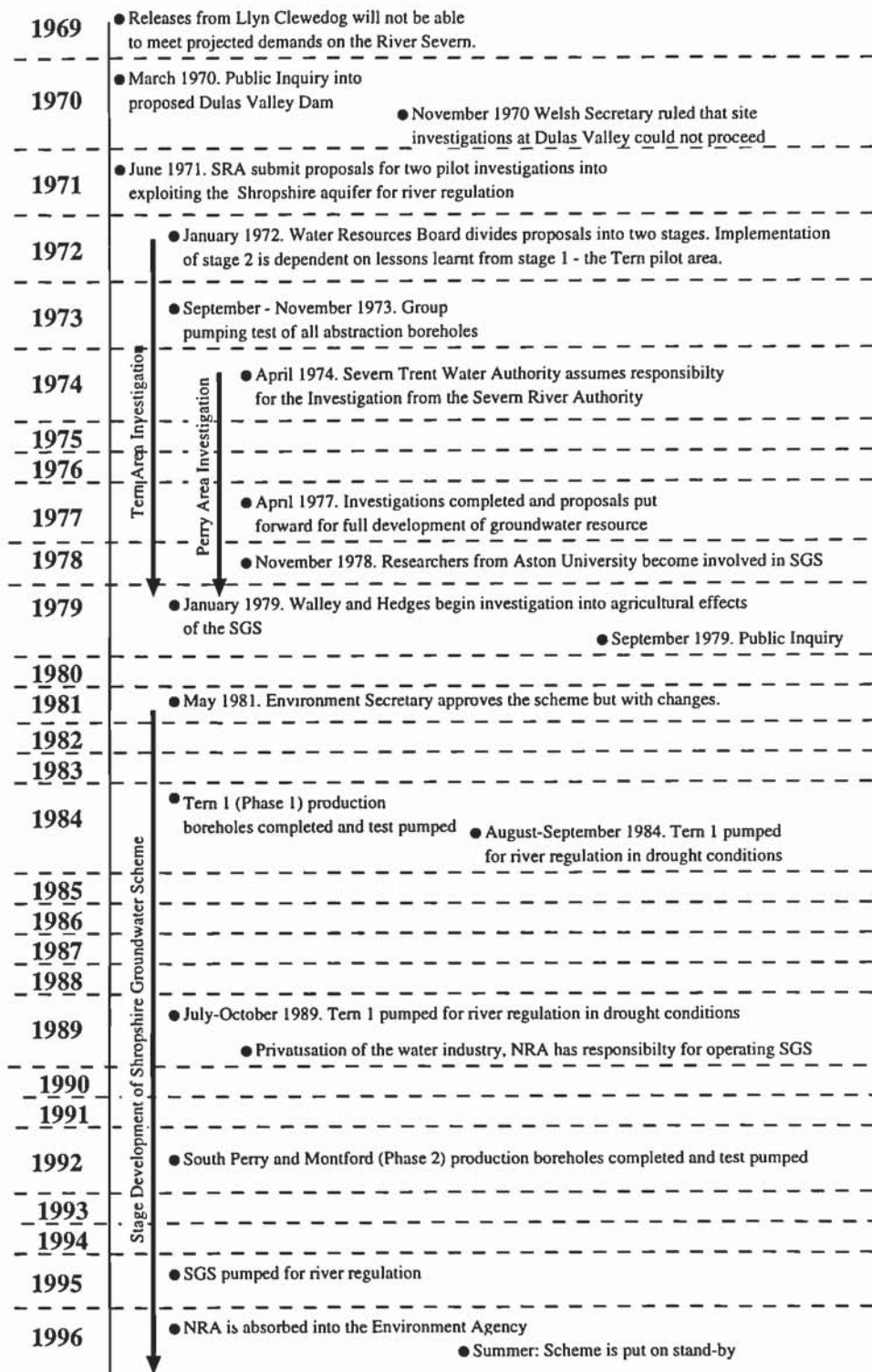


Figure 2.3: Significant events in the history of the Shropshire Groundwater Scheme



Figure 2.4: Soil moisture theory, after Hedges and Walley (1983).

2.2 Previous work on the SGS at Aston University

Researchers from Aston University first became involved in the SGS in 1978, when the Assistant Director of Operations of the STWA presented a discussion paper about the SGS to the Midlands Branch of the Institution of Civil Engineers (Sharp, 1978). Attending this lecturer were Messrs Hedges and Walley from the Department of Civil Engineering, Aston University.

Walley disagreed with opinions given by Sharp that the SGS would have no effects on agricultural production through the reduction of soil moisture. Subsequently Walley was introduced to farmers in the Shropshire area, and in 1979 a program of investigation was initiated, this is described and results are presented in Hedges (1989).

Walley and Hedges (1979) submitted a report to the sponsoring farmers and this formed the basis of his evidence given at the Public Local Inquiry. The report included preliminary results from the field investigation as well as a theoretical treatment of the effect on soil moisture of watertable drawdown. Later laboratory work was performed to validate the theoretical arguments (Hedges, 1989).

2.2.1 Theoretical effects of drawdown on soil moisture

The theoretical argument that Walley (1979) submitted to the Public Inquiry was expanded in Hedges and Walley (1983) and Hedges (1989). The reasoning goes as follows.

The unsaturated hydraulic conductivity of soil is closely related to the soil moisture content. However, the relationship is not linear (aside from hysteresis effects) and there is a rapid change from high to low conductivity over a small soil moisture content range, as illustrated by Figure 2.4.a. Hedges and Walley (1983) stated that;

“...the form of the conductivity/tension relationship, and in particular the tension



Figure 2.5: Movement of the soil moisture profile with watertable drawdown, after Hedges and Walley (1983).

(S_o) at which conductivity falls rapidly is the principal factor governing the extent to which groundwater drawdown can affect soil moisture content. ”

If a soil profile drains without surface inputs or outputs, it tends towards a state in which tension is directly proportional to height above the watertable. But as the tension reaches the critical tension (S_o) at which soil conductivity drops rapidly the movement of moisture becomes negligible. Thus the theoretical profile is never reached, Figure 2.4.b. Hedges and Walley introduced a concept of critical height (h_o); the height above the watertable below which the watertable influences soil moisture. The moisture profile achieves a profile they call the “field capacity profile” (which also includes an element of capillary rise), Figure 2.4.c, however they stress that this is only a theoretical profile:

“ Theoretically it should be possible to construct the field capacity profile for any particular soil by combining the moisture content/tension curve with the tension/height above watertable relationship. However, unsaturated conductivity is difficult to measure, and so little is known of the conductivity/tension relationship for British soils, that this is not possible in practice. ”

However the field capacity profile does give some insight into the way soil moisture will respond to watertable drawdown. When the watertable is lowered by pumping, the soil moisture profile will tend to track it down. Any plants whose roots previously penetrated the zone of the critical height or the capillary fringe will experience a drop in available soil moisture.

Hedges and Walley also considered hysteresis effects. The soil moisture tension/curve will be very different under wetting conditions to that under drying. The tension at any given moisture content on a drying curve will be greater than on a wetting curve. Hence there will only be a

small recovery of the soil moisture within the original critical height if the watertable moves back to its original position.

Soil moisture theory and modelling is reviewed further in Section 6.1.2.

2.2.2 Field work

In 1979 Walley and Hedges undertook field studies in the Tern Pilot area on behalf of concerned farmers. The field studies performed before the Inquiry were necessarily constrained by time. It was not possible for Walley and Hedges to isolate the effects of drawdown from the other soil moisture influence such as; evapotranspiration, interflow, deep percolation and rainfall.

The method they adopted was to use existing boreholes sunk for the pilot area investigation. Soil moisture access tubes were installed at different distances from the chosen abstraction boreholes and the change in soil moisture was monitored with a neutron probe. The cone of depression was used in an attempt to assess the effects of different drawdowns on soil moisture, which were compared against the results of the traditional mass balance method of estimating changes in soil moisture.

Three sites were select for the field study. Each had different groundwater conditions and soil profiles. In summary the results from the pumping indicated:

- at the Greenfields site soil moisture decreases were due entirely to evaporation. The layer of clay drift effectively isolating the soil moisture from watertable drawdown;
- at Heath House where the groundwater is in hydraulic continuity with the soil a significant loss of soil moisture due to pumping was observed from three of the five soil moisture tubes (see site plan in Figure 6.16);
- at Childs Ercall, where the watertable was artesian and the sandy clay soil is underlain by boulder clay, the conditions changed "from slow upward flow under artesian pressure to slow drainage under gravity". Hence loss was slow but continued after pumping ceased and, though there was eventually recovery, this too was slow.

Hedges and Walley divided the responses to watertable drawdown into four classes.

- Watertable 2m or more below the bottom of the root zone. The largest critical height was 1.9m observed at Heath House. Therefore if the initial watertable is 2m or more below the bottom of the root zone no roots can penetrate the critical height zone and experience a loss of water.
- Watertable within 2m of the root zone but isolated from the effects of drawdown by an impermeable layer.

- Watertable within 2m of the root zone and effectively continuous with it. The Heath House observations showed “that a relatively small drawdown in the watertable can remove substantial quantities of soil moisture normally available to crops, providing their roots penetrate into the zone of critical height”.
- When the groundwater conditions are artesian the soil is effectively isolated from the watertable movement by the impermeable layer. However, when the impermeable layer is ‘leaky’ upward flow may contribute to soil moisture, a contribution which may be lost with the drop in watertable.

From this work Hedges and Walley developed a simple rule for identifying vulnerable soils. In areas where there is no impermeable layer a soil is vulnerable if:

$$D < h_o + d_r \quad (2.1)$$

where:

- D — is the depth to groundwater table (L);
- h_o — is the critical height (L);
- d_r — is the rooting depth (L).

2.2.3 Laboratory studies

Critical height (h_o) is dependant on soil type. In order to identify sensitive soils Hedges (1989) undertook a program of laboratory investigations from 1983 to 1989, on real Tern area soils and artificial soils similar to those found in the Tern area. In summary the investigations included:

- standard procedures for determining specific gravity, particle size analysis and bulk density;
- saturated conductivity was measured on undisturbed samples using a constant head permeameter;
- Soil moisture profile investigations using soil columns.

Hedges specially designed soil columns for his experiments. The soils were saturated from the bottom up and then allowed to drain until the field capacity profile was assumed to have been reached. At this point the soil column was stripped and divided into segments (the column had been design for this purpose), and the soil moisture content of each segment was found using an oven drying method.

The purpose of these experiments was to establish a relationship between the critical height (h_o) and measurable soil parameters. Also needed to define the field capacity profile were the saturated moisture content (θ_s), the residual soil moisture content (θ_r) and the height of the capillary fringe (h_c), Figure 2.6. The rate of drainage was also deemed important.

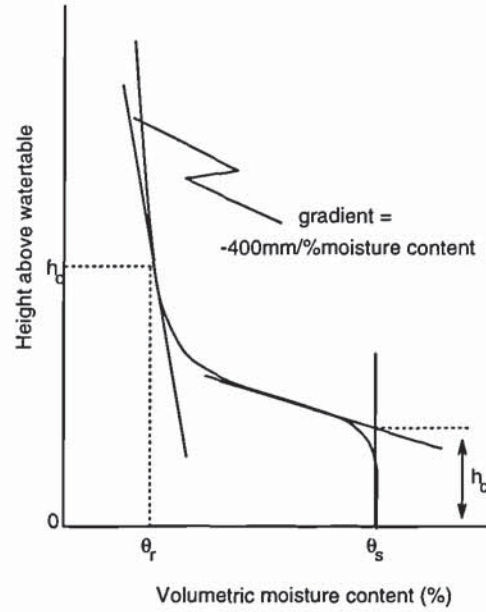


Figure 2.6: Idealised soil moisture profile showing significant factors; θ_s — the saturated moisture content, θ_r — the residual moisture content, h_o — the critical height, and h_c — the height of the capillary fringe.

From the soil moisture column data Hedges generated several relationships using regression analysis.

$$h_o = 58.3S - 7.89Sd + 1350 \quad (2.2)$$

$$h_c = 34.7S + 13.9Sd - 1200 \quad (2.3)$$

$$\theta_r = 1.13C - 0.077Sd + 11.6 \quad (2.4)$$

$$\theta_s = 33.5\% \quad (2.5)$$

where:

h_o — critical height (mm);

h_c — capillary rise (mm);

θ_r — residual soil moisture content %;

θ_s — saturated soil moisture content %;

S — percentage silt;

Sd — percentage sand;

C — percentage clay.

Hedges found that there was only a weak relationship between saturated soil moisture content θ_s and any of the measurable soil moisture properties and, as the standard deviation was low (only 2.15%) Hedges adopted a constant value of 33.5%.

Hedges also showed that hysteresis did indeed prevent a full restoration of soil moisture on the recovery of the watertable, though the effect of this was not as severe as had been anticipated. The question whether the rate of drainage is too slow for soils to be affected by drawdown was not fully resolved;

Hedges cautioned against the uncritical use of his relationships and points out that;

- they have been derived from soil with uniform soil properties down the profile;
- the bulk densities only varied from 1.5 to 1.6 gm/cm^3 , and so are not representative of all soils;
- special caution must be taken when sandy loam soils are considered;
- further study is needed in respect to moisture removal rate.

2.2.4 A technique for environmental impact assessment

Hedges (1989) developed a technique for delineating the areas sensitive to groundwater drawdown and for evaluating the effects of drawdown on crops and trees. The technique involved assembling all the significant parameters (soil type, depth to groundwater, distribution of drift, etc.) in map form and combining the data to identify and delineated the sensitive areas, by applying the rule expressed by Equation 2.1 to areas where there is no impermeable clay layer.

Initially this combination of parameters was done manually, with the distribution and intensity of each parameter being drawn on transparent map sheets. When the sheets were overlaid the cumulative effects of the different parameters could be visually assessed.

Subsequently the process was computerised, each parameter was represented within the computer as a two-dimensional array. To overlay maps it was just a question of mathematically combining corresponding elements in different arrays to produce a new array — the composite map.

The computer used for this was the BBC B computer which had 32Kb of memory (an extra 20Kb was acquired later). The memory restrictions constrained the size of the arrays that could be used. However, even with this constraint the suite of programs that was produced — with programs for data capture, for displaying data, for manipulating data, and for producing hard copy maps — can be seen as an early Geographic Information System (GIS), Section 3.3. The method of map overlay is fundamentally the same as local operations in cartographic modelling (Section 3.3.3). The system produced is described in detail in Hedges (1989).

In order to delineate vulnerable areas, maps of soil type (from which critical height is derived, Equation 2.2), crop distribution (giving root depth), distribution of clay drift and depth to

watertable were combined using Equation 2.1. In order to delineate areas actually affected and evaluate crop loss, maps of different drawdown scenarios were combined with the vulnerable areas map, and a map of the 1980 cropping pattern of the Tern area.

2.2.5 Assessing total soil moisture removal

Hedges (1989) used the empirical expression:

$$\theta = (\theta_s - \theta_r) \frac{a}{(a + h^b)} + \theta_r \quad (2.6)$$

where:

- θ — moisture content at height h above the watertable (%);
- θ_s — saturated soil moisture content of the soil (%);
- θ_r — residual soil moisture content of the soil (%);
- h — height above the watertable (m);
- a — an empirical parameter;
- b — an empirical parameter;

as a basis for quantifying the moisture in the soil profile. From the laboratory work (section 2.2.3) Hedges derived the following expressions for a and b .

$$a = 0.0173S + 0.291BD - 0.294 \quad (2.7)$$

$$b = 2.32 - 0.651 \ln(a) \quad (2.8)$$

where:

- S — percentage silt;
- BD — bulk density (gm/cc).

2.2.6 Drawdown response time

Hedges (1989) developed a series of regression equations from his soil column experiments to describe the drawdown response time (DRT_m), the time in days for a proportion m of the drainable soil moisture to be lost.

$$DRT_{50} = 1.90BD + 0.016C - 2.91 \quad (2.9)$$

$$DRT_{75} = 5.67BD + 0.18C - 9.0 \quad (2.10)$$

$$DRT_{90} = 12.7BD + C - 20.8 \quad (2.11)$$

$$DRT_{95} = 28.6BD + 1.62C - 44.5 \quad (2.12)$$

$$DRT_{99} = 55.2BD + 2.62C - 78.6 \quad (2.13)$$

where:

DRT_m — time for the proportion m of the drainable water to be drained from the soil profile (days);

BD — bulk density (gm/cm);

C — percentage clay (%).

Hedges used these very simple equations to help calculate how long the crops had experienced a soil moisture deficit and how intense that deficit was (Section 4.5).

2.2.7 Assessing crop loss

Hedges (1989) used data obtained from the Gleadthorpe Experimental Husbandry Farm (EHF) to establish relationships between soil moisture loss and yield loss. The Gleadthorpe data (see e.g. MAFF, 1977) concerns the gain in yield with the application of irrigation. Hedges made the assumption that the data could be used to establish relationships between soil moisture loss and crop yield.

“If the yield after drawdown were equated with the ‘without’ irrigation situation, then the no drawdown case could be regarded as the ‘with’.”

Hedges obtained unpublished detailed data from Gleadthorpe (EHF) and used linear regression analysis to derived relationships for different crops.

$$\Delta YD_s = 0.0025MD - 7.71 \quad (2.14)$$

$$\Delta YD_c = 0.18MD - 104 \quad (2.15)$$

$$\Delta YD_p = 0.0017MD - 0.116 \quad (2.16)$$

where:

ΔYD_s — yield deficit of sugarbeet (tonnes/ha);

ΔYD_c — yield deficit of cereals (kg/ha);

ΔYD_p — yield deficit of potatoes (tonnes/ha);

MD — moisture debt ($mm \cdot days$), the integral of the moisture deficit against time graph.

Hedges also developed an expression for loss of milk yield, using various sources.

$$\Delta YD_d = 15.3ML \quad (2.17)$$

where:

ΔYD_d — is the loss of milk yield (litre/ha);

ML — is the total moisture loss from root zone (litre/ha).

2.3 Evaluation

There were several drawbacks or problems with Hedges's (1989) analysis in evaluating with and without pumping scenarios. Hedges uses two snapshots of the watertable position, one at the start and one at the end of the summer. However, in reality the watertable position will vary continuously, and any resulting soil moisture loss will tend to lag behind this movement. This is further complicated in that there are natural fluctuations in watertable position, which can be expected to be particularly marked in a dry year. Plants respond differently to soil moisture loss at different times in their life-cycle, thus the timing of the pumping and the development of the soil moisture deficit is crucial. On a practical note, to perform this kind of analysis over a large area, tens of km^2 , requires a lot of data and makes considerable demands on computer resources, as Hedges found.

Recognising these issues Hedges (1989) made a number of general recommendations for future research needs at the end of his project;

- the EIA technique (Section 2.2.4) should be adapted for use on the current generation of PC's, and should be developed to be as portable as possible;
- the EIA technique should be incorporated into a GIS in order to simplify data management;
- links from the GIS to groundwater and soil-water-plant models, would provide a better representation of reality, and have a potential for commercial exploitation.

This current project is intended to develop Hedges's (1989) work along these lines. The SGS, and the River Tern Basin in particular, remains as a focus but the project is principally academic in nature.

Chapter 3

GIS and process models

The underlying theme of this thesis, in one form or another, is environmental modelling. This theme pervades each chapter, sometimes explicitly — as in Chapters 5 and 6 where computer models of groundwater and soil moisture systems are described — often implicitly — as in Chapter 5 where, as well as explicitly using a groundwater model encoded for a computer, it is also necessary to develop conceptual models to simplify and understand the arrangement of geological structures before the groundwater model is developed.

This chapter is intended to explore the subject of environmental modelling. Firstly the subject is dealt with in general but focusing upon environmental process models, secondly Geographic Information Systems will be discussed, thirdly the integration of GIS and process models will be examined in general and finally the strategy for linking models adopted for this project will be outlined.

3.1 Models

Models are a fundamental part of scientific investigation, and crop up in all branches of science, consequently the literature is vast and will not be reviewed here. For a philosophic treatment of models see Hempel (1965), and Doucet and Sloep (1992) gives a more accessible overview of the theory of modelling.

A working definition might be:

A model is a system or object that resembles some other system or object in a useful way, so that from the behaviour of the model, understanding or prediction of the real system or object is obtained. The model might be a physical or mechanical system or object or a mathematical equation or a set of rules. The resemblance is not perfect, if it were the model would be a copy, the resemblance need only be good enough for the specific purpose of the model.

From this definition it follows that many things can be classed as models; maps (whether

paper or digital), physical laws, empirical relationships, statistically and geostatistically derived relationships and any combination of these. Hence the statement that the underlying theme of this thesis is environmental modelling. A model may be constructed for a variety of reasons, but loosely these can be classified as gaining knowledge and understanding or prediction.

Because a model is not a perfect copy of the system certain assumptions are made (explicitly or implicitly) that allow the model to be simpler than reality. Whatever assumptions are made, they must not compromise the assumption that the model is fit for the task.

An important distinction is between those models which are physically based and those which are not. Physically based models simulate the system by manipulating the inputs in a way which emulates or is analogous to the actual physical processes operating. Other models may successfully simulate the significant features of a system but without emulating the physical process; for example the regression relationships defined by Hedges (1989) (see Chapter 2, Equations 2.2 to 2.17).

3.2 Environmental process models

This section concentrates on environmental process models. For the purpose of this thesis environmental process models will be taken to be that class of models concerned with the flow of matter and energy through natural or man made environments but ignoring physical or mechanical models. They are physically based models and they can include stochastic components. Generally environmental process models include the temporal dimension unless some simplifying assumption is made that allows the system to be treated as steady-state.

Perhaps the simplest models are lumped conceptual models. The term *conceptual model* is used differently by different authors. There are two distinct meanings. Firstly it is used to describe those models where relationships between components of the model are described qualitatively rather than using strict mathematical relationships and rules. Thus a conceptual model of geology might represent the geology as homogenous units abutting at infinitely thin boundaries. The construction of these conceptual models is often the first stage in a more rigorous modelling exercise. The conceptual model of geology, which might only ever be expressed as a geological interpretation in the mind of a geologist might be used to guide the construction of a groundwater model. This class of conceptual model includes models that are not necessarily process models.

In the discipline of hydrology the term *conceptual model* has a more specific meaning. Shaw (1993) describes this as:

“ In conceptual modelling, the catchment processes are described mathematically (e.g. by an equation for evaporation or routing procedure for overland flow), and the storages are considered as reservoirs, for which water budgets are kept. ”

Figure 3.1 shows a simple conceptual model of infiltration.

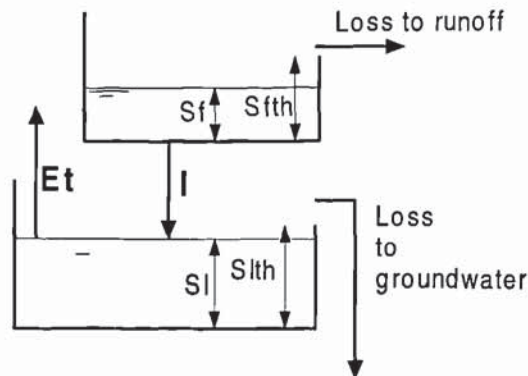


Figure 3.1: A simple hydrological conceptual model representing infiltration as two stores, S_f — surface storage, S_{fth} — surface storage threshold, S_l — amount in soil storage, S_{lth} — soil storage threshold, E_t — evapotranspiration loss rate dependent on soil storage, I — infiltration rate dependant on surface storage.

3.2.1 Lumped and distributed models

Environmental processes vary in space as well as time. A simplifying assumption that is often made is that these spatially varying processes can be adequately modelled by assuming that they occur at a point. Thus the model in Figure 3.1 could be taken to represent infiltration over an entire catchment, say.

If this assumption is not justified for a particular purpose, then a distributed model might be used. The area of interest would be divided into regions and the model applied to each region. The division of the area may be on physically definable units (e.g. sub-catchments or hill-slopes in hydrology, soil types, vegetation units or climatic zones in ecological modelling), or into arbitrary units (e.g. grid squares). Rather than a distinct divide between lumped and distributed models, the situation should be seen as a spectrum, distributed models are usually lumped at the sub-catchment or grid square level.

Distributed models may offer increased accuracy but they have received some criticism (Beven, 1989) both in the theory and in the practice of their use. In practice they generally require much more data than lumped models. It is often the case that when data is lacking each element in the distributed model is given the same value for a particular parameter. The process of calibration can be criticised; when one has thousands of input parameters distributed in space and time and a single output signal, such as with a distributed runoff model where the output is a flood hydrograph, the exercise can become one of curve fitting and the physical basis of the model is compromised — the model achieves the right answer for the wrong reasons. Even when a model is very finely distributed, very rarely are the model cells at the same scale as that at which field

measurements are taken. Beven (1989) warned there was a lack of a theory from which to integrate sub grid scale measurements to the effective values required at the grid scale.

3.2.2 Practical models

As well as maintaining the underlying assumption of fitness, a model must be practical if it is to be used. That is to say the cost of using it, whether measured financially, in effort, time or computation, must not be excessive in relation to the project that it is being applied to. Hanks and Hill (1980) defines a practical model as:

“ ... one that has been developed to a point that practical questions can be answered to some degree of reliability. ”

Over the last 20-30 years and increasingly with the increasing power and availability of computers, models have been encoded into computer programs. With the advent of generally available powerful computers, the limits of what would previously have been considered excessive have increased but this consideration is still holds. So what makes a model practical for use? Broadly the three criteria are;

- it must model all subsystems within the scope of the model adequately;
- data requirements must be realistically achievable;
- must not take too long to perform the modelling.

The first consideration has interesting implications. On examining computer implementations of models it is often the case that they are not one model but several models that have been brought together in a single package. Some of these sub-models may be strictly physically based and others less so. Taking the computer model SWMS_2D as an example (see Section 6), unsaturated flow is modelled according to Richard's Equations, however there are sub-models, such as those that describe how water content varies with tension or how conductivity varies with water content, which have been developed using a mix of physical theory and empirical methods (Simunek, Vogel and van Genuchten, 1994).

Most of the issues of practicality arise from the complexity of the model. So to say a model must be fit for the task is only the first criteria, the most appropriate model will be one that is only just fit for the task. EPA (1992) describes a useful set of selection criteria for model code.

- Reliability; the model should have undergone some kind of peer review, there should be evidence of use, it should have been compared with field studies.
- Usability; the availability of binary, source, pre and post processors, data resources, standardised data formats, user manuals, sample problems, the necessary hardware, transportability, user support.

- The trade off between model performance (speed, accuracy etc.) with the resources required to use it (staffing, data, hardware) should be examined and justified.
- Public domain or proprietary; if not public domain, then the use of proprietary code should be justified.
- The objectives; if the objectives have to be changed to allow for the capabilities of the model these should be justified.
- The model assumptions; the theoretical basis of the model should be analysed.
- Modifications; any modifications should be treated as rigorously as the original code.

This is perforce a very brief synopsis of the subject of modelling in general and environmental process modelling in particular. However, it has introduced terms and concepts used throughout this thesis.

3.3 GIS

The topic of GIS has been included in this chapter because the discipline of GIS is the discipline of creating and manipulating models which describe the world. Burrough (1986) defines GIS as:

“...a powerful set of tools for collecting, storing, retrieving at will, transforming, and displaying spatial data from the real world for a particular set of purposes. This set of tools constitutes a ‘Geographical Information System... ”

An alternative definition might be:

a system for managing and integrating models of different aspects of the world. As such a GIS has; (a.) internal models e.g. data models, such as the raster or vector models (see Section 3.3.1) which describe how data about the environment should be structured in the GIS database for storage, retrieval and manipulation; (b.) functions which exploit different models to manipulate the data e.g. different interpolating functions, such as geostatistical kriging, distance weighted or spline based, use different models to predict the way attributes vary through space; and (c.) an understanding of external models that allow the importing or exporting of data e.g. NTF standard¹ and the communication with outside entities that have models that differ again, e.g. other IS or GIS, other software packages and automatic loggers etc., Figure 3.2.

Under this definition GIS have yet to reach maturity, as the understanding of external environmental process models will be seen to be less than satisfactory.

There is a very large literature base on the subject of information systems (IS) in general (Avison, 1992; Senn, 1990). A recurring theme in this literature is that an IS is *not* just technology, but has an organisational aspect - that is to say an IS includes the definition of procedures and

¹ NTF is the National Transfer Format used by the Ordnance Survey for supplying digital data, see OS (1993)

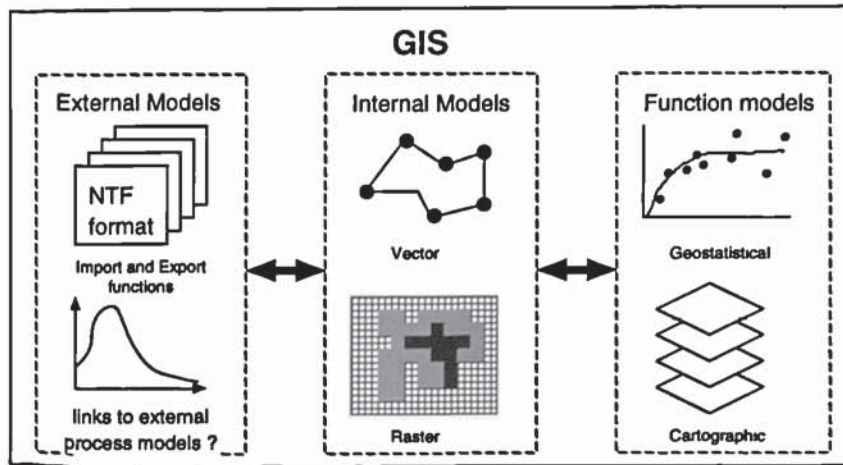


Figure 3.2: GIS: a system for managing and integrating different models of the world. The GIS stores data using internal models, to communicate with outside entities, such as process models, or other data formats the GIS must also *understand* the models that these outside entities use and know how to translate them into its own internal models. Similarly many functions also use alternative models of the world to achieve some useful manipulation of the data, the GIS must again have the understanding to translate between these models and the internal models (this understanding is generally encoded into translating and resampling functions, here represented by the thick black arrows).

assignment of responsibilities for such activities as accessing, updating, backing up, disseminating and analysing the data. Thus an IS need not necessarily include computer technology.

GIS in large organisations have a similar organisational aspect as IS in general, see Charnock, Snoxell and Elgy (1994). However GIS inevitably involve computer technology. The term GIS will be used for a software package or set of software tools for handling geographic data. The term GIS database will be used for the internal data upon which the GIS acts.

The remainder of this section which will briefly discuss the data-structures which form a significant part of the discussion in this thesis, the typical functionality of a GIS, cartographic modelling which might be describe as the fundamental analysis operation of a GIS and the issues of data quality, accuracy and error.

3.3.1 Data-models and data-structures

To represent the complexities of the real world in a computer which at the lowest level stores data as a one dimensional list of 0's and 1's, a series of abstractions are made, Figure 3.3 (Peuquet, 1984). The first abstraction from reality is the data-model and the second is the data-structure. The third is the file-structure², which is the province of Operating System (OS) developers and program-

² File-structure in this case means the representation of the data on the physical storage medium (usually a magnetic disk), and not the file format, as might be specified by a supplier of data which is a data-structure.

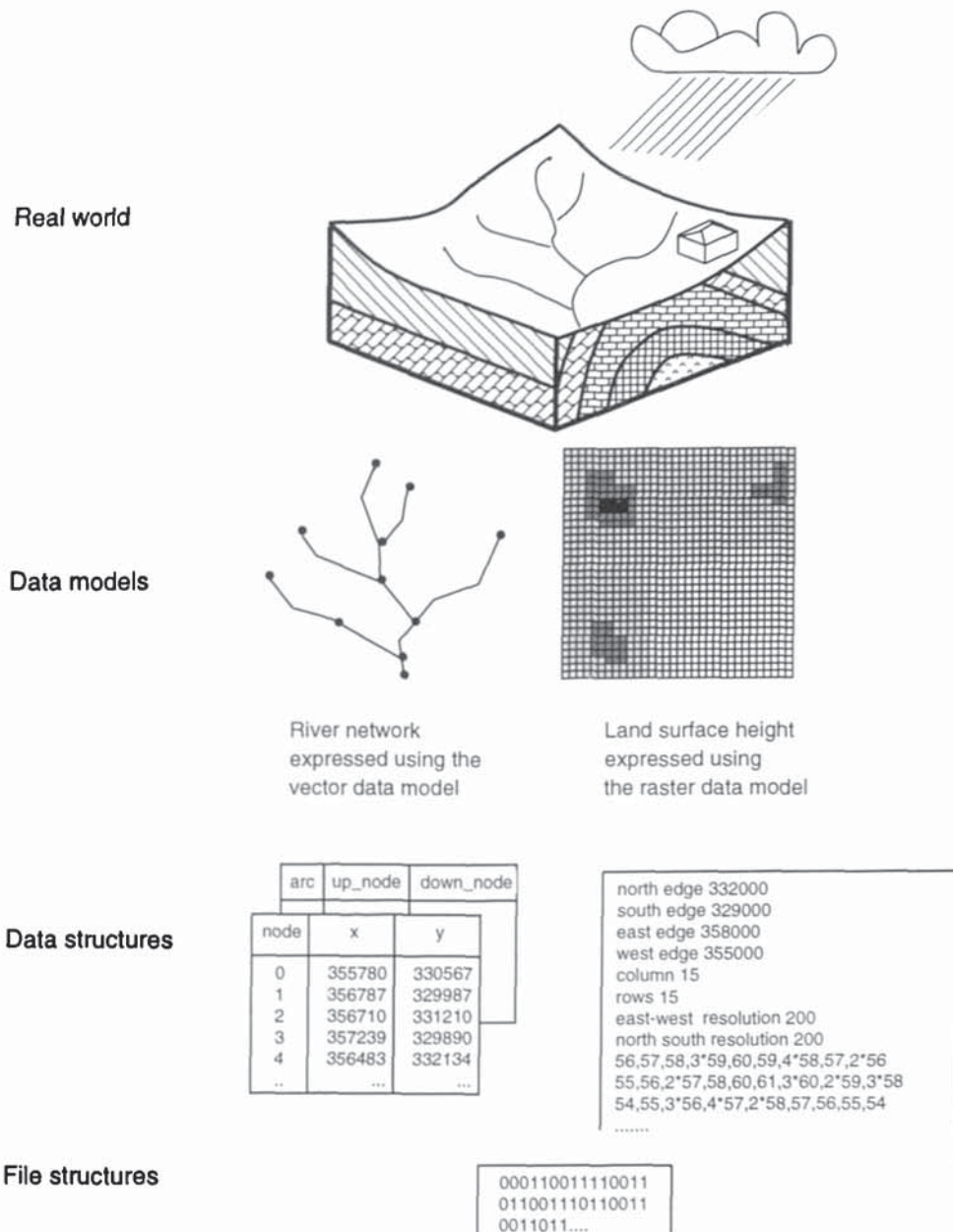


Figure 3.3: GIS: abstraction from reality.

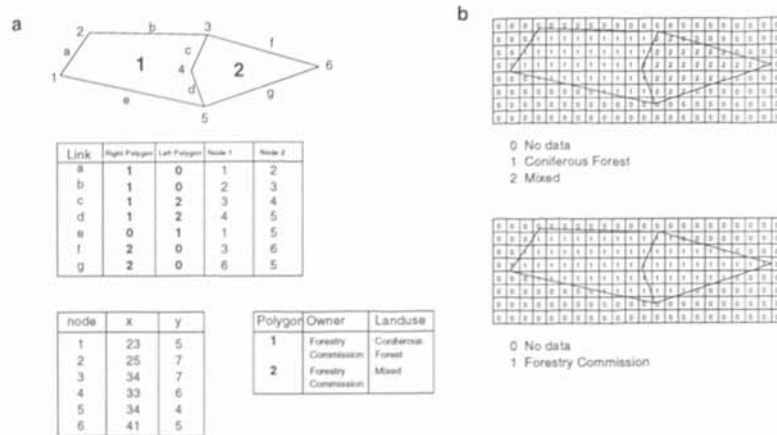


Figure 3.4: (a) The vector or object data-model and a typical vector data-structure and (b) the raster or field data-model and typical raster data-structure.

language compiler writers and is not considered in any depth here.

The most commonly used data models in GIS are the raster model or field model and the vector model or object model. Figure 3.4 illustrates the difference between these two approaches. The vector model populates a co-ordinate space with geometric objects, each of which has associated attributes. The raster models divides the landscape into fields each region having an associated attribute.

Data-models and structures are not confined to GIS of course, the development of data-models is a major preoccupation in database engineering and computing in general. The computer implementations of process models discussed above, have data-models to guide the structure of the input data, the internal structures of data upon which the algorithms are applied and the output format of the data. These data-models are usually derived from the mathematical or conceptual formulation of the model, for example a finite difference or element grid or as a series of reservoirs and connections.

3.3.2 GIS functionality

Maguire and Dangermond (1991) present a systematic approach to the functionality of GIS, the tasks a GIS should perform. They point out that the data-model has significant influence on this functionality. The classes of operations are;

- data capture;
- data transfer;
- validation and editing;
- storing and structuring;
- restructuring and generalisation;
- transformation;
- query;
- analysis;
- presentation.

Query and analysis are probably the most important functions and Maguire and Dangermond argue that it is these operations that differentiate GIS from other computer systems. Query operations relate to inventory and asset management, analysis is concerned with assessing the effects of various scenarios. It is not intended to explore the whole range of query and analysis operations here. Just one class of operation is taken as an example, one that is particularly useful to a project involving the use of remotely sensed data and other raster images — cartographic modelling.

3.3.3 Cartographic modelling

Tomlin (1991) describes cartographic modelling as:

“ ... an algebra in which single-factor maps are treated as variables that can be flexibly manipulated using a small but highly integrated set of cartographic functions. ”

Cartographic modelling is yet another kind of environmental modelling that is included in this thesis. It is arguably the fundamental analysis methodology of GIS, though it has a longer history going back to the manual overlaying of maps. In cartographic modelling the underlying assumption of the model is that the world can be represented as spatially varying layers of attributes, and where some combination of these attributes will derive some spatially varying resultant attribute. For example a very simple model of vulnerability to soil erosion might be envisaged as the coincidence of slope greater than a threshold and vegetation sparser than some threshold, Figure 3.5.

Hedges's (1989) procedure for identifying vulnerable areas is another example of cartographic modelling, see Section 2.2.4. Burrough (1986) classifies cartographic modelling operations into three different classes (Figure 3.6, NB cartographic modelling is not restricted to raster GIS).

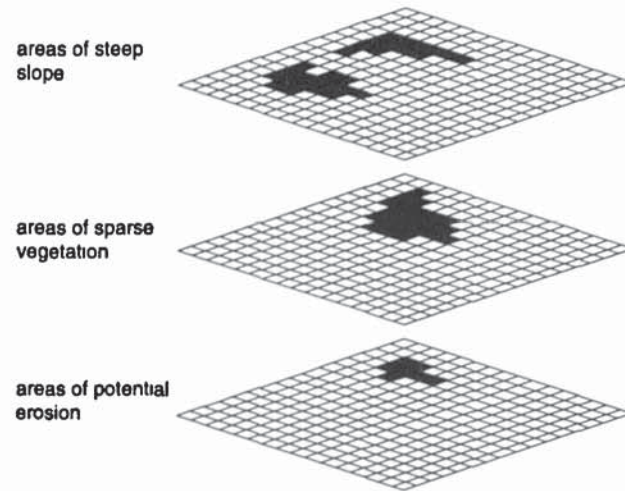


Figure 3.5: A simple overlay operation, brings together maps of slope and vegetation to delineate areas of erosion.

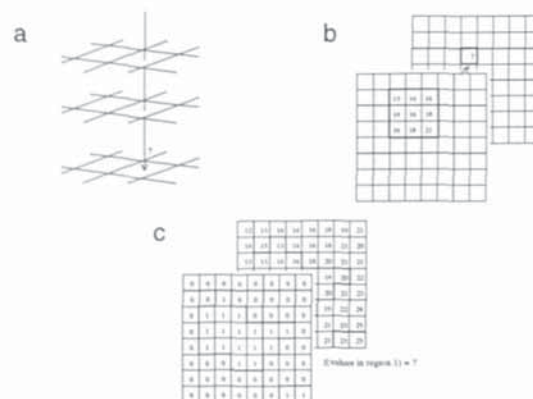


Figure 3.6: Operations in cartographic modelling. (a) a point operation, (b) a neighbourhood operation and (c) a region operation.

- Point operations, the same spatial location on different attribute maps are combined (e.g. as in the overlay example above). These operations include arithmetic operations (such as adding, subtracting, etc.) and Boolean operations (e.g. *and*'s and *or*'s).
- Neighbourhood operations, a new attribute is produced at a location by combining and manipulating values at neighbouring locations, these operations include the filters commonly used in image processing, also slope and aspect operations.
- Region operations, a region on a map (e.g. the forest area in Figure 3.4) is given an attribute in the basis of some combination and manipulation of all locations in that region.

By combining operators cartographic models can be very sophisticated and powerful. Examples of cartographic models crop up throughout this project.

3.3.4 Data quality, accuracy and error

Central to any database is the data itself. The world is complex, it is not possible for any model to represent the world completely; how well the data represents the world is an important consideration. A commonly used term is data quality, Clark (1993) pointed that there is no satisfactory definition of data quality, it embodies aspects of; accuracy, precision, reliability, documentation and lineage.

Another important consideration is the application. Data adequate for one application may not be adequate for another. An obvious example is data digitised from maps at a particular scale, this data should only be applied with care to projects involving larger scales.

Error and accuracy are, of course, major considerations in many fields including remote sensing and hydrologic modelling. Both these disciplines have large bodies of literature concerning this area. It is considered here as the GIS literature represents the most systematic study of the issues of spatial data quality, and because error from the hydrological models, from the remote sensed data and from all other sources will combine within the GIS. Various authors have considered sources of error and have attempted to classify them, Thapa (1992) gives a good review.

3.3.5 The subject of GIS

GIS is a huge subject and arguably a discipline in its own right. Certainly it has spawned many Master's courses, one or two undergraduate courses, professional associations and several academic and commercial journals. No attempt has been made to comprehensively review the subject here, for further reference the reader is referred to Laurini and Thompson (1995) and especially Burrough (1986)

3.4 GIS and environmental process modelling

In recent years the technology of GIS has matured and its use in hydrological projects has expanded. The linkage of GIS to hydrological models and to process models in general is an obvious development, and many recent papers have given both practical examples or dealt with the issues of this. It is clear from the literature that there are many different approaches to linking single and multiple models to a GIS and that an important component of each linkage is translating between the different representation of reality used by the GIS and the process models

3.4.1 Advantages of linking GIS and environmental process models

Currently there is a massive effort to join the technologies and techniques of GIS and of environmental modelling. At the recent HydroGIS 96 conference (Kovar and Nachtnebel, 1996) approximately 30 out of the 83 papers presented dealt with aspects of linking hydrological models with GIS, and the NCGIA conference "Third International Conference/Workshop on Integrating GIS and Environmental Modelling" was, of course, largely concerned with this subject (Goodchild, Parks and Steyaert, 1996).

But why is there this surge to link these two technologies? The answer is because they complement each other, since a large proportion of the effort expended when using a model is in assembling and structuring the data. For example, the model SWMS_2D (see Section 6.4) uses simple text files as a means of interfacing with the user, Figure 3.7. The user specifies the inputs in accordance with a strict format. To manually construct these or to alter them during model refinement and calibration is an laborious, error prone process. As the models are made increasingly complex, the inputs can become geometrically larger and more unwieldy.

Organisations and companies have responded to this by embedding existing models in a pre-post-processing packages. For example, SWMS_2D is now marketed as the model HYDRUS which includes a windows based interface (Kool and van Genuchten, 1991), the groundwater model Modflow (see Section 5.4) has several pre- post-processing packages written for it such as Chiang and Kinzelbach (1992). This approach can cause problems; unless vast expense is spent on the package then the pre- post-processor will only have a small subset of functions available to a full-blown GIS. Also, if many different models and their attendant processors are used then there will be a significant overlap in functionality between all the different processors, a waste of effort on the part of the developers and on the users, who must learn several different methods of doing the same thing. Again if several such models are used then there is either a great deal of moving data around, restructuring and resampling and all the attendant dangers that that involves, or else there is duplication of data with all the dangers that that entails, e.g. inconsistencies between different versions of the same dataset.

By linking models into a GIS, it is the data which is placed at the heart of the Information System and not the model(s), Figure 3.4.1. Models can be added or made redundant, but the


```

*** BLOCK A BASIC INFORMATION *****
Heading
'Test site one'
LUnit TUnit MUnit (indicated units are obligatory for all input data)
'm' 'day' '-'
Kat (0 horizontal plane, 1 axisymmetric vertical flow, 2 vertical plane)
2
Maxit TolTh TolH (max number of iterations and tolerances)
200 0.005 0.05
lWat lChem CheckF ShortF FluxF AtainF SeepF FreeD DrainF
t f t t t t t t t t t t
*** BLOCK B MATERIAL INFORMATION *****
MMat Mlay hTab1 hTabM NPar
1 1 00001 1000 9
thr ths tha thm alfa n Ks Kx thk
0148 350 0148 35 0.92 1 2766 0.06 0.6 36
*** BLOCK C TIME INFORMATION *****
dt dtMin dtMax DHul DHul2 MPL
001 1e-4 1 0 1 3 3 6
TPrint 1,TPrint 2,TPrint(NPL) (print-time array)
20 40 60 80 1
*** BLOCK D SINK INFORMATION *****
P0 P2H P2L P3 r2H r2L
-0.10 -2.00 -8.00 -80.00 0.005 0.001
POptm 1,POptm 2,POptm(NMat)
25 -25
*** END OF INPUT FILE 'SELECTOR IN' *****

*** BLOCK E MODAL INFORMATION *****
NumMP NumEl IJ NumSP NOba
122 60 2 4 0
n Code x x h Conc Q N B Arr Bxx Bzz
1 -4 0.00 0.00 -7.80 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00
2 -4 1.00 0.00 -7.80 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00
3 0 0.00 -0.1000 -4.27 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00
4 0 1.00 -0.1000 -4.27 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00
5 0 0.00 -0.2 -0.74 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
6 0 1.00 -0.2000 -0.74 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
7 0 0.00 -0.3000 -2.22 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
8 0 1.00 -0.3 -2.22 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00

116 0 1.00 -5.7000 1.70 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
117 0 0.00 -5.8000 1.8 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
118 0 1.00 -5.8 1.80 0 E+00 0.00E+00 1 0.00 1.00 1.00 1.00
119 0 0.00 -5.9000 1.90 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00
120 0 1.00 -5.9000 1.90 0 E+00 8 E+00 1 0.00 1.00 1.00 1.00
121 3 0.00 -8.0000 2.0 0.00E+00 0 E+00 1 0.00 1.00 1.00 1.00
122 3 1.00 -8.0000 2.0 0 E+00 0 E+00 1 0.00 1.00 1.00 1.00

*** BLOCK I ELEMENT INFORMATION *****
e 1 j k l Angle Anis1 Anis2 LayNum
1 1 3 4 2 0.00 1.00 1.00 1
2 3 5 8 4 0.00 1.00 1.00 1
3 5 7 8 8 0.00 1.00 1.00 1
4 7 9 10 8 0.00 1.00 1.00 1
5 9 11 12 10 0.00 1.00 1.00 1
6 11 13 14 12 0.00 1.00 1.00 1

55 109 111 112 110 0.00 1.00 1.00 1
56 111 113 114 112 0.00 1.00 1.00 1
57 113 115 116 114 0.00 1.00 1.00 1
58 115 117 118 116 0.00 1.00 1.00 1
59 117 119 120 118 0.00 1.00 1.00 1
60 119 121 122 120 0.00 1.00 1.00 1

*** BLOCK J BOUNDARY GEOMETRY INFORMATION *****
Node number array
1 2 121 122
Width array
0.50 0.50 0.50 0.50
Length
1.00
*** END OF INPUT FILE 'GRID IN' *****

*** BLOCK I ATMOSPHERIC INFORMATION *****
*** Greenfields 1980 *****
SinkF qGVLF
f t
GVLFOL Agh Bgh (if qGVLF=t then Agh=Bgh=0)
0 -0.001687 -0.0002674
tlimit Maxal Maxal = number of atmospheric data-records
0 36
hCritS (max allowed pressure head at the soil surface)
1e30
tAtm Prec Prec rSoil rRoot hCritA rt ht crt cht
0.425 0.00272 0 0 0.0 1000000 0 2.00 0 0
7.708 0.00055 0 0 0.0 1000000 0 2.20 0 0
11.5 0.0002 0 0 0.0 1000000 0 2.20 0 0
13.48 0.00409 0 0 0.0 1000000 0 2.40 0 0
14.56 0 0 0 0.0 1000000 0 2.60 0 0
14.81 0 0 0 0.0 1000000 0 2.85 0 0
14.87 0.01225 0 0 0.0 1000000 0 2.97 0 0
15.56 0 0 0 0.0 1000000 0 2.68 0 0
18.5 0 0 0 0.0 1000000 0 2.875 0 0
20.48 0.00476 0 0 0.0 1000000 0 2.88 0 0
22.825 0.00346 0 0 0.0 1000000 0 -0.02 0 0
25.54 0.00477 0 0 0.0 1000000 0 -0.015 0 0
27.47 0.00198 0 0 0.0 1000000 0 -0.035 0 0
29.54 0.00684 0 0 0.0 1000000 0 -0.055 0 0
32.55 0.00241 0 0 0.0 1000000 0 -0.08 0 0
34.54 0.00467 0 0 0.0 1000000 0 -0.26 0 0
36.53 0.0001 0 0 0.0 1000000 0 -0.14 0 0
39.52 0.00322 0 0 0.0 1000000 0 -0.105 0 0
43.46 0.00006 0 0 0.0 1000000 0 -0.13 0 0
47.5 0.00923 0 0 0.0 1000000 0 -0.15 0 0
50.47 0 0 0 0.0 1000000 0 -0.32 0 0
54.64 0.00058 0 0 0.0 1000000 0 -0.16 0 0
57.55 0.01158 0 0 0.0 1000000 0 -0.2 0 0
61.54 0 0 0 0.0 1000000 0 -0.24 0 0
64.47 0 0 0 0.0 1000000 0 -0.2 0 0
68.60 0 0 0 0.0 1000000 0 -0.2 0 0
68.61 0 0 0 0.0 1000000 0 -0.2 0 0
71.77 0.00004 0 0 0.0 1000000 0 2.2 0 0
81.54 0.00058 0 0 0.0 1000000 0 2.48 0 0
96.48 0.00180 0 0 0.0 1000000 0 2.47 0 0
142.48 0.00142 0 0 0.0 1000000 0 2.62 0 0
153.48 0.00144 0 0 0.0 1000000 0 2.7 0 0
162.48 0.00129 0 0 0.0 1000000 0 2.67 0 0
181.49 0.00290 0 0 0.0 1000000 0 2.67 0 0
205.48 0.00384 0 0 0.0 1000000 0 2.63 0 0
232.48 0 0 0 0.0 1000000 0 2.67 0 0
*** END OF INPUT FILE 'ATMOSP IN' *****

```

Figure 3.7: Sample input files for the SWMS_2D model.

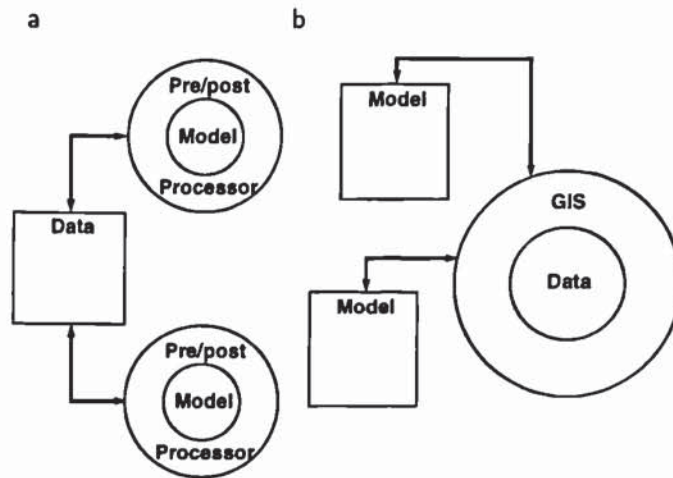


Figure 3.8: Data handling approaches for existing numerical models, (a) the pre/post-processor approach, (b) the GIS approach.

data stays the same in the format of the GIS.

So the reasons for the current move to link GIS to environmental models is largely for the mundane reason of data administration convenience. However, the importance of this should not be underestimated, as time and time again organisations are finding that it is the data and the data handling that make up one of the most significant costs in any modelling effort. So the whole is greater than the sum of its parts — whilst the models allow very sophisticated analysis, the GIS enables very powerful data handling facilities.

Wilson (1996) was concerned that GIS had not brought anything new to environmental modelling (he noted “new analysis methods for routing water flow across the landscape” as an exception to this). But this is to miss the point. What GIS should bring environmental modelling is not innovation but amenity. Model developers (that is the person who constructs the conceptual model in the most abstract sense, whether they then chose to encode it on a super computer or write it on the back of an envelope) should be given as much freedom as possible to use imaginative data-models and concepts to model whatever it is they are concerned with. They should not be tied into a GIS and into a particular way of structuring and thinking about the world that this entails. It is the task of the package developers, whether of a full blown GIS or a pre-/post-processing packages, to implement the model as code and include all the subordinate facilities of data capture, manipulation, restructuring etc. that turn the innovative new model into something practical. The two tasks of model developer and package developer should not be confused.

3.4.2 Approaches to linking GIS and environmental process models

Fedra (1993) described various levels of integration from the very simple situation where the GIS is used for the creation of model input and the analysis of model output, to a closely integrated single system which:

“ ...would merge the two approaches, such that the model becomes one of the analytical functions of the GIS, or the GIS becomes yet another option to generate additional state and output variables in the model, and to provide additional display options... ”

Fedra presented a conceptual design of such a system and stressed the importance of user interfaces, and the use of expert systems or knowledge bases. The Hydra Decision Support System is an example of this approach, and Ireland (1995) describes it as;

“ ...a crop irrigation decision support system (DSS) ... that is intended for water utilities, farmers and regional and local government in Mediterranean countries ... although not an expert system per se, hard coding of the latest scientific knowledge endows it with an unrivalled knowledge base ”

Hydra links soil moisture and crop models together with embedded functions from the GRASS GIS. The programming language Tcl is used along with its graphical user interface extension TK to construct the user interface. There is a considerable investment of effort in constructing the user interface; essential given the number of different types of users targeted. Development to a finished product took 3 years with 25 people working on the project at any one time and the project had 750 000 ECU's of funding (Ireland, 1995).

Elgy, Maksimović and Prodanovic (1993) recognised that a viable link between existing urban drainage models and GIS could be realised by using small inexpensive gluing routines.

Harris, Gupta, Woodside and Ziemba (1993) linked the CFEST (Coupled Fluid, Energy and Solute Transport) model with the GIS Arc-Info, for a large groundwater basin investigation. The development was a continuous process parallel with the actual groundwater investigation, so their approach was distinctly different to that adopted by the developers of Hydra:

“ Instead of building a single new environment, the two (Arc-Info and CFEST) have been left apart, with all their capabilities left intact, but tied together with a network of programs that communicate between them... ”

Hydra and the systems of Elgy et al. (1993) and Harris et al. (1993) represent opposite ends of a spectrum of approaches to system design. At one extreme, e.g. Hydra, a total modelling environment is constructed, models are reimplemented as new code, often by combining relationships from several different models to simulate only those environmental subsystems of interest. Considerable effort may be expended on constructing the user interface and compiling expert systems or knowledge bases. The advantages are that a complete, possibly commercial, product is produced, that is easy for a non-expert to operate. However, such systems can incur considerable cost, in time, investment and maintenance, as Fedra (1993) points out:

“ There is a trade-off between efficiency and ease of use and the flexibility of the system. The more options are predetermined and available from a menu of choices, the more defaults are provided, the easier it becomes to use a system for an increasingly smaller set of tasks” ”

The other extreme, e.g. Elgy et al. (1993) or Harris et al. (1993), involves linking existing model codes and GIS with communicating or translating programs, and can be viewed as developing a “quick and dirty system”. This can mean a considerable savings in time and expense. Against this is the requirement for expertise on the part of the user, and the reliance on standard GIS functions to be adequate to the tasks of data handling and display.

Both extremes are valid and have advantages and disadvantages. There is, of course, considerable scope to follow some middle path. As Harris et al. (1993) point out, given enough development, the links between the model and the GIS can be made, “seamless and transparent to the day-to-day user”. In effect Fedra’s (1993) aim of utilising the models as “analytical functions of the GIS” is realised with considerable less effort than rewriting the model internally in the GIS.

The choice of approach should be a response to such considerations as: end user expertise, the size of the user base, resources, time schedules, and the importance or worth of the decisions being made. If the model is to be used by a small number of expert users then a sophisticated integrated package with a long development time is not justified. If the user base is small and inexpert then the decision to supply a modelling system within the capabilities of the user must be based on the worth of the decisions to be made, the cost of making a wrong decision and the decrease in the risk of a wrong decision that the system will bring.

3.4.3 Achieving linkage

Closely related to the subject of differing approaches to linkage is the question how is the linkage to be achieved? Whichever approach is adopted it must be achievable with the software tools available to hand. Three general situations can be identified; unsupported linkage; GIS supported linkage and operating system supported linkage.

Unsupported linkage

Provided a complete programming language is available, it is possible to link any piece of software to any other. However, if such a linkage is not envisaged by the designers of the software then this can be a very tricky and time consuming process. Two basic approaches can be used, the first is to attempt to work out the file structures of the package. With a bit of experience combined with trial and error, it may be possible to decipher the internal data structures of the package it then becomes possible to read and write to these structures without going through the user interface of the package³.

³There are legal considerations to this approach if the file structures are commercial property.

Alternatively, if the packages have exporting and importing routines that write internal data structures to and from some common external standard structures which may be public domain or at least published (and GIS generally do have such functions), then the process of making a link is far simpler. One need only embed these porting functions within some higher level process and use the common structures as the means of communication.

Which ever approach is taken it is possible with more or less effort to make the linkage transparent to the general user. Though this may be time consuming and the resultant package may be very inefficient.

GIS supported linkage

All GIS have the capability of exporting and importing data, indeed this may be taken as a defining criteria of GIS, so some form of unsupported linkage, as described above, is always possible. Most GIS designers have gone further than this and expect some form of linkage or at least customisation of the GIS by the user. Thus Arc-Info has the full programming language AML which can be used to customise the Arc-Info interface and to structure and store complex database queries. So, in Arc-Info, AML could be used either to extract the data and write it into a form for an external model, or write the output of the model back into the Arc-Info database. With some effort a model itself could be implemented with AML⁴. However the system is implemented, AML would be used to build the user interface to the model. Thus it is the GIS that provides all the facilities for making the link.

Some developers have taken a different route. For example in Idrisi there is no specific language, rather the developers have documented the important data structures and suggested a protocol to follow in the expectation that users will write their own modules. The developers of GRASS have gone further than this. Because GRASS is public domain there are no commercial considerations so the developers have made available not only the GRASS executable code, but also the GRASS source code and the program libraries used in development. A large proportion of the GRASS functionality comes from external application developers following the GRASS protocol.

Kopp (1996) stated:

“ The integration of GIS and modelling has evolved being GIS-centric. This is mainly the result of the GIS being able to perform so many of the tasks for the modeller; data manager, pre and post-processor, display. With emergence of UNIX and X⁵, GIS software became not only a geodata server and geoprocessing tool box, but also a graphical user interface builder and application development environment. If we needed some other tools for statistics or 3D visualisation, we either linked other software packages to the GIS or wrote a data conversion program to move data between packages. GIS provided a common ground on which people and their data interact. ”

⁴ AML is a macro language, written for ease of use, it is probably not suitable for writing a computationally intense model, though some have done this.

⁵ X is a standard for a windows operating system for UNIX machines

Kopp is describing what is referred to above as GIS supported linkage — the GIS developers rather than the OS developers had assumed responsibility for enabling linkage.

However GIS, even if following the Arc-Info approach, only support linkage to environmental process models in a limited way. Tools for building the interface for structuring the queries necessary for extracting the required data from the GIS may become very sophisticated. But could there be more explicit support for linking process models which despite their many differences also share many similarities, and could the process of linking be more automated? This discussion is taken up in Chapter 8.

Operating system supported linkage

Data processing usually requires performing several different operations to data. When data processing operation are *ad hoc*, that is when the operations that need to be performed can not be predicted before hand it is often the case that no single package will perform the required task. So the data must be moved from software package to software package. In the most simple case the data is imported into a package, the operation is performed and then the data is exported. In addition some manual editing of the data may be performed before it is imported into the next package. If the data set is large this can be cumbersome. Operating system developers have tried to support the passing of data between application, but how can this be done when it is not known what sort of data and what sort of packages the user might be using? The simple answer to this is that developers of applications follow protocols and standards laid down by the operating system developers.

The approach adopted by the UNIX system has been to use *shells*, *pipes* and *redirection* (see Appendix A.1.2). With this approach information is passed between applications as *streams* of text. For example:

```
more file_name | grep word > new_file_name
```

This command reads a file using *more*, the stream or text is “piped” using the symbol “|” to the command *grep*. This command selects lines from the file if they match the *word*. The “redirection” symbol > redirects the stream of text generated by *grep* to a file *new_file_name*. It is the *shell* that interprets the command and marshals the resources of the operating system to accomplish the task. Over the years several different UNIX shells have been developed, but they have similar syntax and features. Perhaps the most important feature of the shells is that they are fully functional programming languages. The above command could be typed in at the keyboard as a *command line* statement. Alternatively it could be written into a text file which could be executed repeatedly. The statement could also be embedded into any of the standard programming structures; i.e. loops, selections etc.

Some applications require the data to be more structured than a simple stream of text. However the UNIX protocol does not specify any higher level structures. So it is up to the application

developers to use appropriate sequences of characters to structure the data. Thus, for example, it is common to use a “,” to delimit distinct numbers in a stream of text.

What if different application developers use different structures and different character sequences to indicate these structures? In response to this problem various tools have been developed to manipulate text files and streams of text, the most commonly used of these tools are *awk* (Aho, Kernighan and Wienberger, 1988), *sed* (Bourne, 1983) a stream editor or PERL (Wall, Christiansen and Schwartz, 1996). For example, this is a typical use of *sed*:

```
more file_name | sed -e 's/,/\\/g' > new_file_name
```

This command reads a file of the form:

```
xcoord,ycoord,description new_line
xcoord,ycoord,description new_line
etc...
```

into the form:

```
xcoord|ycoord|description new_line
xcoord|ycoord|description new_line
etc...
```

this happens to be the way GRASS structures site or point data.

This section has merely touched the surface of this subject and much more sophisticated manipulation than the example above can be performed with *sed*, more so with *awk* and PERL is a fully functional programming language. Examples of all of these tools (sometimes one embedded within another) are scattered throughout the thesis.

A new paradigm has been developed in recent years, that might be termed the *windows* approach (as it is typified by the windows operating systems of the software companies Apple Macintosh and Microsoft).

The main and most obvious feature is that the operating system, as well as mediating between applications and system resources, also provides a standard graphic interface for the applications. This kind of interface has become the norm for modern computers, to the extent that X windows has been developed to give UNIX a similar *feel*, however the interface will not be discussed here.

What is important for this discussion is that the windows environment provides a completely different way in which applications can be linked. Kopp (1996) stated that these developments would *revolutionise* the linkage of GIS and modelling. This revolution will be based not on;

“ ... major advances in hydrology or GIS, but on evolution of industry standards and development tools that enable software from these and many other technologies to interact more easily. ”

Moving data between applications within the windows OS is simple. Most users are aware of the idea of a clipboard. A piece of formatted text, or picture is copied from one application on

to a clipboard and then *pasted* onto another. This requires direct user involvement however, the windows approach also allows one application to communicate with another and *ask* for the data that it requires (e.g. Microsoft's OLE, Object Linking and Embedding, standard.). For this to be achieved the applications must understand each other, there must still be protocols but these are much more sophisticated than the UNIX paradigm described above.

The sort of data that is passed between applications includes formatted text, graphics and database records, but as yet the protocols do not explicitly include Geographically Referenced data.

“ Emerging trends in data access, inter-application communication, component based software and development environments are changing GIS from being the core of a spatial decision support system, or the glue that holds it all together, into GIS being just one component of an integrated custom application. ”

In essence Kopp (1996) is predicting a move away from GIS supported linkage to operating system supported linkage. As this happens the distinction between the loose linked and the closely integrated approaches identified above will become more blurred.

3.5 System Design

The task at hand is to build a modelling system which will support simulation of the important physically processes within the Shropshire Groundwater scheme.

3.5.1 General Strategy

As discussed in Section 3.4.2, the general strategy of linking should be a response to such considerations as: end user expertise, the size of the user base, resources, time schedules, and the importance or worth of the decisions being made. Despite the exercise being a real life example, the project is academic. The author has designed the system as well as being the person to use it, so there is no need to worry overmuch about the user interface etc., budgets are low and time is constrained. In this case the loosely linked approach is appropriate.

The strategy for system development adopted here might be characterised as the prototyping approach. The principal behind the prototyping approach when applied to information systems is that a working system is produced that the user can experiment with and comment on, and so provide the basis of a design for the next prototype thus prototyping is an iterative process. However for this project there was only time to work on the first prototype. Senn (1990) listed the main features of the prototyping approach.

- Speed of development, not efficiency of the prototype performance, is the overriding concern of both system analyst and end-user.
- The initial prototype is likely to be incomplete or unsatisfactory in one or more ways. Changes in specification and modification of the system are expected.

- Users should use the system in a hands-on fashion to determine by trial and error the changes and enhancements that are desirable.
- Each iteration will result in one or more of the following changes:
 - modification of the data used in processing or the manner in which data are stored in the system;
 - changes in existing features;
 - addition of new features.
- A typical prototyping experience will have four to six iterations.

It will be seen from this list that the prototyping approach to information system development and the loosely linked approach to GIS and model integration are compatible.

In Section 3.4.3 three alternatives were outlined for linking GIS and models; unsupported linkage, GIS supported linkage and operating supported linkage. As is stated in Section 3.5.2 the GIS GRASS which was chosen for this project.

The GRASS developers have made available programming libraries and documentation of structures in the expectation that external users will develop their own applications (see Appendix A.1). Thus for the bulk of the linkage a GIS supported approach is adopted. However GRASS has been written very much in accordance with the UNIX paradigm, the vast majority of its commands have a non-interactive command-line version which can be embedded within shell scripts. During the learning process that accompanied developing the links it was found that some of the linking was better achieved using an operating system supported approach. The end result is a mixture of the two approaches, not elegant as a system, but fulfilling the discovery aims of a first prototype.

Figure 3.9 (a data flow diagram) shows how the approach will work, a data flow diagram is a standard way of specifying an information system in terms of the movement of information missing out the distraction of considerations such as hardware. It shows processes that act upon the data (the circles or bubbles), data stores (horizontal lines), data flows (arrows) and entities outside the system (the boxes). Figure 3.9 is a generic diagram illustrating the way each model is linked into the GIS.

So, following the data-flow around the system, the user interacts with top-left bubble to specify the dataset, i.e. which object in the GIS corresponds with which item within the model structure. Thus if this were a groundwater model the user might identify which raster holds the transmissivity data for a particular layer in the model. The dataset becomes itself an object within the database, something that can be stored, recalled, edited, copied and deleted. When the user chooses to run the model, the top-right bubble is activated, it finds the dataset within the GIS database, uses that to identify which objects in the database are needed, those objects are then retrieved and written into the model input format. The model input files are not part of the GIS database, they are created temporarily when the model is run and then removed. The third process bubble is the model itself, which is called to operate on the data input files and create some output files.

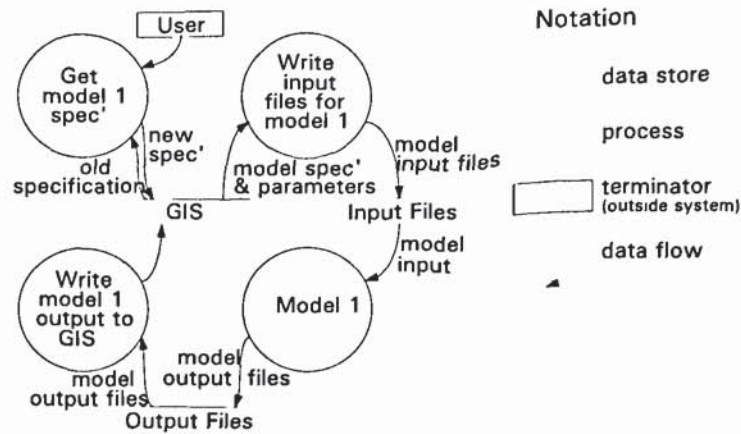


Figure 3.9: Data flow diagram showing a logical way of linking models and GIS via input files.

Finally the last process is used to write the required output from the output files into GIS database objects.

Once a model is linked in it can then be used along with all the other models of the system. Communication between the GIS and the models is via the data files, so there is a need for a controlling process to call the components in the correct order. Figure 3.10 illustrates the system as a hierarchy of control in a structure diagram — another commonly used way of specifying an Information System — it shows the passing of control between the GIS, the models and the controlling process. The controlling process could be the user interacting directly with the system components or a shell script which specifies the order of operation.

Figure 3.11 illustrates how individual models could be combined to model the effect of Shropshire Groundwater Scheme on crop production. In this configuration the groundwater model is used once across the whole region, however the soil moisture and crop models are run at each grid square. Alternative configurations could be achieved using the same components but a different controlling process.

In summary, the general strategy adopted for this research has been to develop a prototype by loosely linking several pre-existing models with a GIS using a mixture of the GIS supported and OS supported approaches. The models required must simulate groundwater, soil moisture and crop growth (Charnock, Hedges and Elgy, 1996).

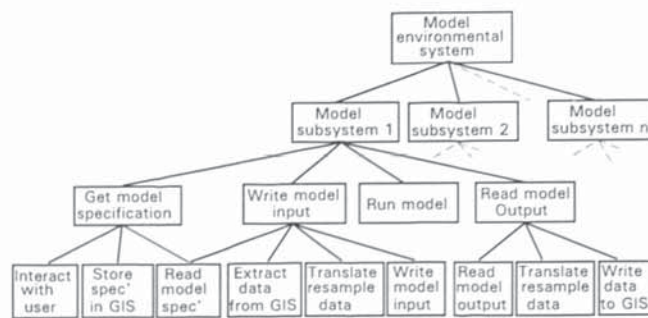


Figure 3.10: Structure diagram showing hierarchy of control of GIS linked numerical models.

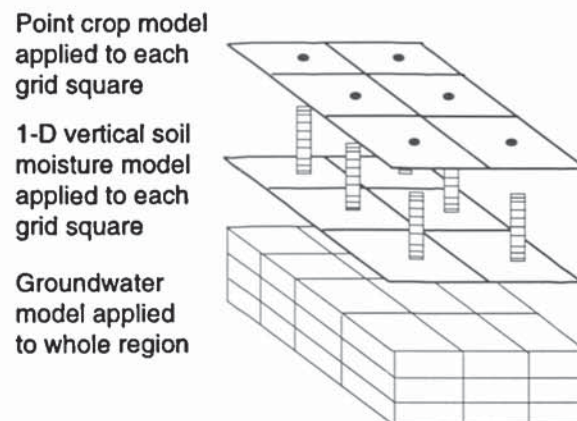


Figure 3.11: A possible configuration of the individual component models. The groundwater model is applied to the whole region, but the crop model and soil moisture model are applied to individual grid squares.

3.5.2 The Geographic Resource Analysis Support System (GRASS)

As briefly stated in the preceding section the GIS chosen for this project was the Geographic Resource Analysis Support System (GRASS), developed by the U.S. Army Construction Engineering Research Laboratory (CERL) (Shapiro, Westervelt, Gerdes, Larson and Brownfield, 1993). It was selected for this project because;

- it is an “open system” (i.e. all the source code is available, and the file formats and libraries are documented);
- it has been placed in the public domain;
- it comes with a large set of libraries with which to construct new applications within the GRASS environment;
- and it was written in C, a programming language the author is familiar with.

GRASS is principally a raster based GIS, see Section 3.3.1, but it has some limited vector capabilities. The user interface of GRASS is very simple — though it does have some applications that exploit “mouse” interaction, it uses for the most part a command-line, question/answer and form based interface. The GRASS shell into which commands are typed is simply an extension of the UNIX shell. The upshot is that GRASS can seem unfriendly to new users and the interface looks primitive compared with modern GIS that have graphical user interfaces (GUI). Alternatively this makes GRASS a very good environment in which to experiment and “try out” new ideas and applications, development can be performed on several levels, from low-level C programming to shell scripting (where existing GRASS and UNIX programs are brought together within a script).

GRASS has been placed in the public domain and thus appears free, however there are costs in time and requirements of expertise that arise through not having the support available with commercial packages. Since the beginning of this project GRASS has been through several upgrades, from GRASS 4.0 to GRASS 4.1.5. with floating-point capabilities. So part of the project has been to compile, test and debug the new versions of GRASS as they are released.

GRASS first appeared some ten to fifteen years ago, it was developed because at that time there was no commercial software appropriate for the needs of army land resource managers. GRASS was placed in the public domain, and since 1985 CERL has released upgrades and documentation as well as provided some technical support. These have been made available to the global community via the Internet.

GRASS began with integer rasters, the original concept was that each pixel held an integer which acted as a key into a category file, the integer itself would have no physical meaning in the real world. Inevitably as people realised the possibilities of such operations as cartographic modelling, the integers were treated as having physical meaning contrary to the original concept of the developers. For a long time this, and the inability to distinguish ‘0’ as a value (e.g. zero

land surface height) from '0' as an indicator of "no-data" or "null" have been major weaknesses within GRASS. These have now been rectified with the release of the "floating point upgrade", which also allowed rasters to hold "null" values.

In the past two years, now that commercial software houses produce systems comparable with GRASS, work on GRASS by CERL has been wound up. CERL has entered into partnership with several companies to ensure continued support for the US Army's GIS needs. So GRASS is moving into a commercial phase, with companies such as Logiciels et Applications Scientifique (LAS) producing commercial versions (LAS's version is called Grasslands). The last release of GRASS by CERL is GRASS4.1.5, with floating point and "null" data capabilities was released as a *beta*⁶ upgrade.

3.5.3 *r.mapcalc*

r.mapcalc is the main program in GRASS for performing cartographic modelling. It is described here because it was used extensively for this project and snippets of *r.mapcalc* scripts are used to illustrate operations throughout this thesis. The basic format of an *r.mapcalc* command is:

```
r.mapcalc new_raster_name=" f(old_raster_name1,old_raster_name2,...)"
```

Which creates a new raster of the size of the current GRASS region⁷. Based on some function of existing rasters. *r.mapcalc* operates on a pixel by pixel basis. The value in each pixel of the new raster is calculated in turn by applying the function to the corresponding pixels in each existing raster.

The function can be a mixture of arithmetic operators:

- + ;
- - ;
- / (i.e. ÷);
- * (i.e. ×);

logical operators:

- && (and);
- || (or);
- == (equals);

⁶*beta software* is the term applied to software when it is first released, it is usable but with the implicit understanding that there are likely to be "bugs"

⁷At any time GRASS has an active region within which all operations are applied, the active or current region has north, south, east and west edge and north-south and east-west resolution. For any GRASS program whenever a raster is used it is "filtered" through this region. The GRASS region is altered with the command *g.region*.

- != (not equals);
- > (greater than);
- < (less than);
- <= (smaller or equal);
- >= (greater or equal);

and functions, such as:

- sin();
- sqrt();
- if().

There is also an offset operator which refers to a pixel relative to the current pixel i.e. `raster[4,3]` which refers to a pixel 4 rows below and 3 columns to the right of the current pixel. A typical *r.mapcalc* command is:

```
r.mapcalc aquifer_top="if(isnull(aquifer) || isnull(drift_type),null(),if(drift_type==3,dem-5,dem))"
```

which is the equivalent of:

```
for each pixel[row x, column y] in current region do:
  if(aquifer[x,y] is null or drift_type[x,y] is null)
    set aquifer_top[x,y] to null
  else if(drift_type[x,y] is clay drift (i.e. type 3) )
    set aquifer_top[x,y] to value of dem[x,y] - 5)
  else set aquifer_top[x,y] to dem[x,y]
```

which is a command for adjusting the position of the confining layer in an aquifer by lowering it an arbitrary 5m in areas of clay drift. For further information about *r.mapcalc* see Westervelt, Gerdes, Larson and Brownfield (1993).

3.6 Evaluation

The idea of “the model” is central to this thesis, the term is probably overused. In this chapter “the model” has been defined, and environmental process models have been looked at. GIS have been briefly discussed and the linkage between GIS and environmental process models has been discussed. Finally, the modelling system design has been outlined and the GIS to be used, GRASS, has been introduced. Figure 3.12 shows the elements and connections of the system which will be described in the next three chapters. Many thousands of lines of code were written for this

project in several languages principally C, PERL, and Bourne shell scripts, examples of some of the programs are included in Appendix A.

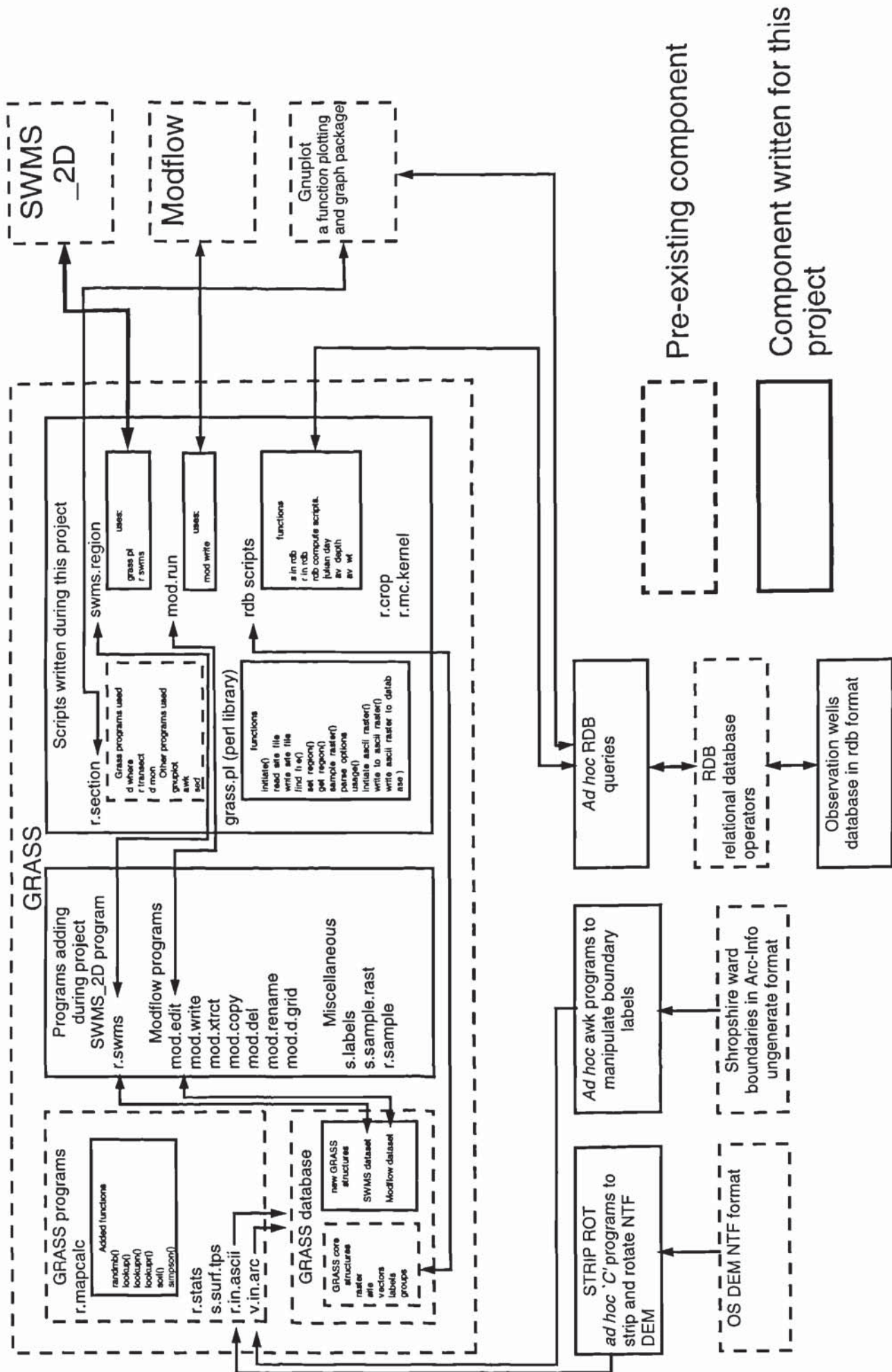


Figure 3.12: Elements of the modelling system — pre-existing and those developed especially.

Chapter 4

Cartographic modelling for EIA of groundwater drawdown

As described in Chapter 2 Hedges (1989) built a set of software tools in order to perform the task of delineating areas vulnerable to groundwater drawdown. As a beginning to the project, and to show that his work was a form of cartographic modelling it was decided to repeat his work within a standard Geographic Information System (GIS). The GIS chosen for this analysis is GRASS (see Section 3.5.2).

4.1 Data

Hedges began the analysis by first assembling the required data. Initially this was the production of paper maps showing the distribution of the attributes of interest, but later on these were converted to a digital form for the computer. The analysis presented here is performed on a UNIX System (specifically a Sun System SPARCstation) which has much more memory than the BBC B used by Hedges, consequently the data can be at a much finer resolution, see Figure 4.1. So although Hedges digital data was available in a convenient computer readable form, it was decided to use his paper maps, and in some cases the original documents, as the source of data for this current analysis.

This decision required a large amount of work, particularly digitising, in order to structure the data in a form that could be used by the GIS.

Because of the severe memory constraints of the BBC computer Hedges could only process a quarter of the region shown in Figure 4.2 at any time. For the purposes of examining his method the same region will be used but, because of the far greater memory capacity of the current computer the region can be analysed whole and at a greater resolution. Hedges used pixel elements of 100×100 m, for this analysis the resolution chosen was 40×40 m.

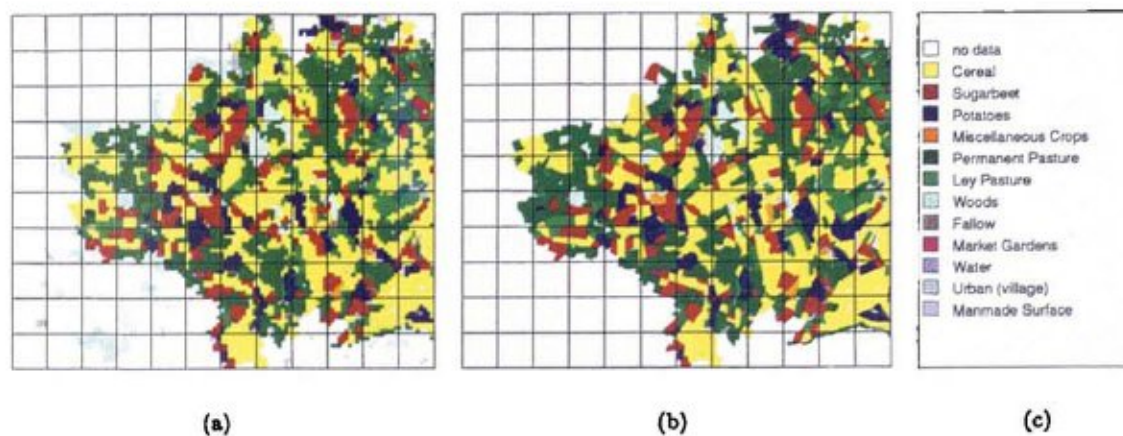


Figure 4.1: The difference in the possible resolution of agricultural maps between (a) a BBC raster and (b) a SUN SPARCstation raster



Figure 4.2: The region used by Hedges (1989).

4.2 Delineating potentially vulnerable areas

The analysis to delineate potentially vulnerable areas simply involves applying the rule expressed by Equation 2.1 to areas where there is no impermeable clay layer.

The basic data required are distribution of soil units and depth to the watertable. Hedges used a very laborious method to map depth to the watertable. He manually drew contours of watertable height on an Ordnance Survey map showing contours of land surface height, then, by visually subtracting levels, divided the area into zones of different depth ranges to the watertable.

An alternative to the above procedure is to use a digital elevation model (DEM) purchased from the Ordnance survey and the height of watertable interpolated (using standard GIS routines) from well records or from groundwater contours drawn by a hydrogeologist. For this analysis depth of watertable contours presented by Skinner (1979) at the Shropshire inquiry were interpolated using the GRASS facility *r.surf.contour* to produce a raster representing a surface of groundwater depth.

Another landscape attribute that acts as a constraint to Equation 2.1 is whether the groundwater within an area is in hydraulic continuity with the soil. Hedges used two variables to define these areas; drift thickness and drift type (e.g. clay, sand etc.). Hedges manually combined the two attributes (drift type represented on a map by discrete areas and thickness represented by contours) according to the rule that hydraulic continuity exists where the drift is not clay or the thickness is less than 5m. This can be achieved within the GIS using *r.surf.contour* to make a surface from the drift thickness contours and *r.mapcalc*:

```
r.mapcalc hyd_cont = "if(drift_thickness<5.0 || drift_type!=3,1,null())"
```

In English the command says “generate a raster with a 1 in each grid square where drift thickness is less than 5m and the drift type is not clay (labelled in the drift map by the number 3)”.

As well as being far faster than manually combining the maps this has the advantage that the rule can quickly be “fine-tuned” should further information become available.

A raster showing critical height was generated by reclassifying a soil map of the region into critical height using the values calculated by Hedges. To delineate the potentially vulnerable areas Equation 2.1 is expressed as an *r.mapcalc* statement where the rooting depth is taken as 2.5m¹ and discounting area not in hydraulic continuity:

```
r.mapcalc "pv2.5 = if((2.5+ho)>depth && hyd_cont,1,null())"
```

In English this query says “select those areas where the depth to watertable is less than the critical height plus 2.5m and in hydraulic continuity with the aquifer” — the results are shown in Figure 4.3. Hedges calculates a total sensitive area of 1927ha, the sensitive area in Figure 4.3 is 3036ha². This is largely due to using a more conservative root depth, however a visual comparison

¹ For this example a root depth of 2.5m is chosen this is greater than any crop root depth found by Hedges (see Table 4.1) and thus is a very conservative assumption.

² Calculated using the GRASS facility *r.report*.

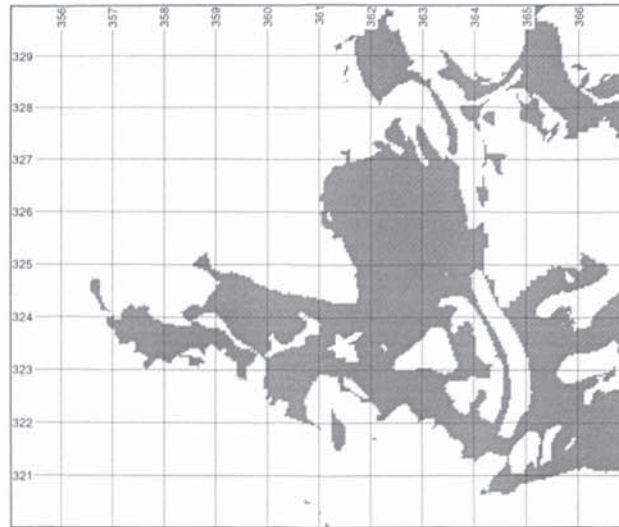


Figure 4.3: The raster *pv2.5* showing potentially vulnerable area, assuming a root depth of 2.5m.

between the output from Hedges shows discrepancies which suggest that Hedges has not followed the technique as stated, making such comparison invalid in any case.

4.3 Delineating actually vulnerable areas

Delineating actually vulnerable areas requires two new sets of data in addition to those used in Section 4.2, one describing the drawdown of the watertable, and one describing the distribution of crops. As a demonstration the typical drawdown presented by Skinner (1979) in his evidence at the Shrewsbury inquiry, is used along with the crop distribution surveyed by Hedges (1989).

This analysis again applies the rule in Equation 2.1, only this time rooting depth will vary with space, and effects will only be felt where there is a drawdown. To get a distribution of root depth over the region of interest the agricultural survey map (Figure 4.1.b) is reclassified into typical root depths. Hedges surveyed the literature to obtain these root depths and his findings are used here (Table 4.1).

At the inquiry Skinner (1979) presented modelled (see Section 5.2.3) drawdown patterns for typical pumping scenarios. Skinner presented this data as contours, so again a raster surface was generated with *r.surf.contour*. Skinner fails to give a 0m contour for his drawdown plot and so, as areas beyond the 0.5m contour (the lowest given) not only have a low drawdown but will experience it later than other areas, and because this is a demonstration of the technique the 0.5m line is taken as the limit of the drawdown. This is an assumption implicitly made by Hedges. As before *r.mapcalc* is used for the query:

```
r.mapcalc "actual_vul=if(drawdown>=0.5 && (ho+rooting.depth)>depth && hyd_cont,1,0)"
```



Illustration removed for copyright restrictions

Table 4.1: Crop rooting depths used for analysis, after Hedges (1989).



Illustration removed for copyright restrictions

Figure 4.4: Actual vulnerable areas, using Skinner's (1979) typical drawdown and Hedges's (1989) crop distribution.

The result of this query are presented in Figure 4.4, again this is similar to the results given by Hedges, though there are differences for the same reasons given in Section 4.2.

4.4 Quantifying moisture loss

Is it possible to use cartographic modelling to quantify the loss of soil moisture within the root zone? This is the next step in quantifying the crop loss for a given pumping scenario.

If Equation 2.6 is the soil moisture at any point above the watertable in the soil profile. Then the total percentage soil moisture (θ_T) between 2 points (h_1 and h_2) in the soil moisture profile is:

$$\theta_T = a(\theta_s - \theta_r) \int_{h_1}^{h_2} \frac{1}{a + h^b} dh \quad (4.1)$$

where:

- θ moisture content at height h above the watertable (%);
- θ_s saturated soil moisture content of the soil (%);
- θ_r residual soil moisture content of the soil (%);
- h height above the watertable (m);
- a an empirical parameter;
- b an empirical parameter.

Rather than attempt to solve this analytically and because the curve produced is a simple “S”-shaped curve, an estimation technique is appropriate and Simpson’s $\frac{1}{3}$ rule was chosen (Chapra and Canale, 1985). So that the soil moisture loss across the landscape could be evaluated, Simpson’s $\frac{1}{3}$ rule was added to *r.mapcalc*. The procedure for adding new functions to *r.mapcalc* is explained in section A.3.1 and the new function is documented in section A.3.4. The syntax of the new function, *soil()*, is as follows;

r.mapcalc “soil_moisture=soil(upper_limit,lower_limit,a,b, θ_s , θ_r ,10)”

where:

- soil_moisture— is the raster to be produced;
- upper_limit is the upper limit of the soil zone of interest, measured positive from the watertable;
- lower_limit is the lower limit of the soil zone of interest, positive from the water table;
- a & b are empirical parameters of Equation 2.6;
- θ_s is saturated soil moisture content as a fraction;
- θ_r is residual soil moisture content.

So to calculate the soil moisture within the zone of intersection between the root zone and the critical height before drawdown the following query might be used. So logic has to be included to account for possible truncating of the critical height by the ground surface or of the root depth by the watertable as illustrated in Figure 4.5.a, b, c & d. the query is then:

```
r.mapcalc soil_moisture_before = "eval( \
upper_limit=if(depth<ho,depth,ho), \
lower_limit=if(depth>2.53,depth-2.5,0), \
sm=if(upper_limit<lower_limit,soil(upper_limit,lower_limit,a,b,theta_s,theta_r),0)"
```

³ Here a constant root depth of 2.5m is assumed, but this can be replaced with a raster describing root depth in a more complex query.

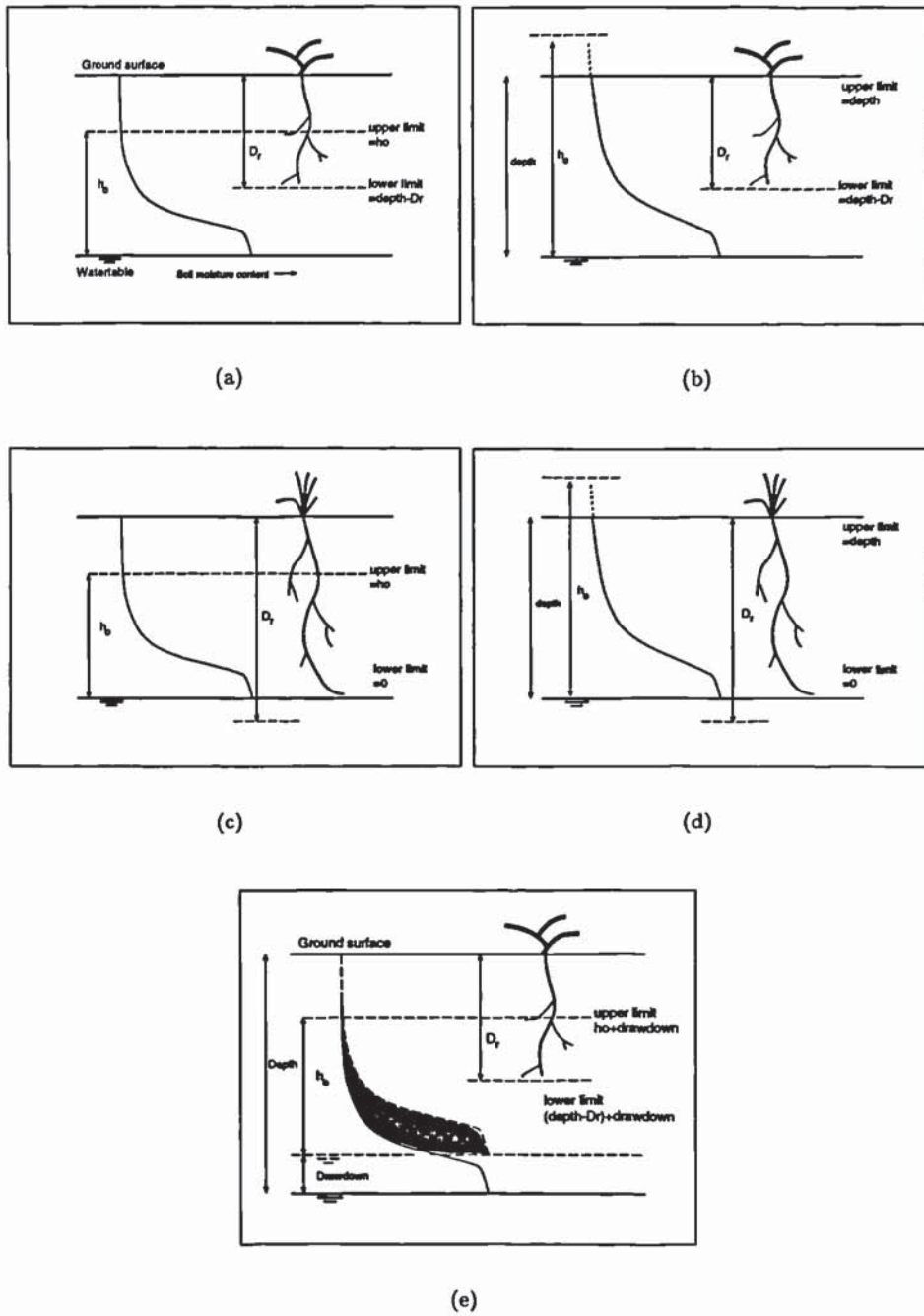


Figure 4.5: The various situations in calculating the moisture loss due to drawdown; (a) the zone of moisture loss is limited by the critical height and the height of the bottom of the root zone above the watertable; (b) the critical height intersects with the ground surface which becomes the upper limit; (c) root growth is constrained by the watertable which becomes the lower limit; (d) combines the effects of b & c; and (e) when calculating the soil moisture in the zone after drawdown the limits are raised by the amount of the drawdown.

To calculate the loss of moisture due to drawdown, i.e. the *after* conditions, the amount of the drawdown is simply added to the limits, and a similar query is used, e.g:

```
r.mapcalc "soil_moisture_after=eval( \
upper_limit=if(depth<ho,depth+draw_down,ho+draw_down), \
lower_limit=if(depth>2.5,0+drawdown,(depth-2.5)+drawdown), \
sm=if(upper_limit>lower_limit,soil(upper_limit,lower_limit,a,b,theta_s,theta_r),0))"
```

The amount lost of course is the difference between the two raster generated by the queries above, e.g:

```
r.mapcalc loss="if(draw_down<=0.5 || hyd_cont==0, /
0,soil_moisture_before-soil_moisture_after)"
```

The results of this final query are shown in Figure 4.6.

4.5 Quantifying yield loss

Having developed a method to quantify the moisture loss, the next stage is to attempt to determine the yield loss. Hedges wrote a BBC program to perform this task, EIACOST. The logic of EIACOST is followed here but using the operations of GRASS.

In Section 2.2.7 relationships between crop yield and moisture loss developed by Hedges (1989) are presented. Taking cereals as an example Equation 2.15 gives a relationship between yield deficit of cereals (ΔYD_c kg/ha) and moisture debt (MD mm.day).

The total soil moisture loss has been established in Section 4.4. Hedges developed a simple rule for establishing the duration of the moisture loss experienced by the crop:

$$D_{ML} = (\text{date of harvest or senescence}) - (\text{pumping start date}) - DRT_{75} \quad (4.2)$$

where:

D_{ML} Duration of moisture loss experienced by the crop (days);

DRT_{75} Time for 75% of the drainable moisture to be drained from the profile (generated using Equation 2.10).

Hedges distinguished between 9 crop types in his study, so to emulate EIACOST with *r.mapcalc* a complex query is required involving at least 9 levels of embedded "if"s. The following statement takes cereals as an example, using the 1974 scenario presented by Skinner (1979):

```
r.mapcalc "yield_loss=if(agriculture == 1,\
eval(duration_of_moisture_loss=2304-1305-(5.678 * B+0.18 * C-9.0)6),\
```

⁴ The 230th day of the year is taken as the day of harvesting cereals.

⁵ Hedges takes the 130th day of the year as the "earliest likely date for the start of pumping".

⁶ The DRT_{75} term, Equation 2.10. A and B rasters describing the spatial variation of parameters from the equation.

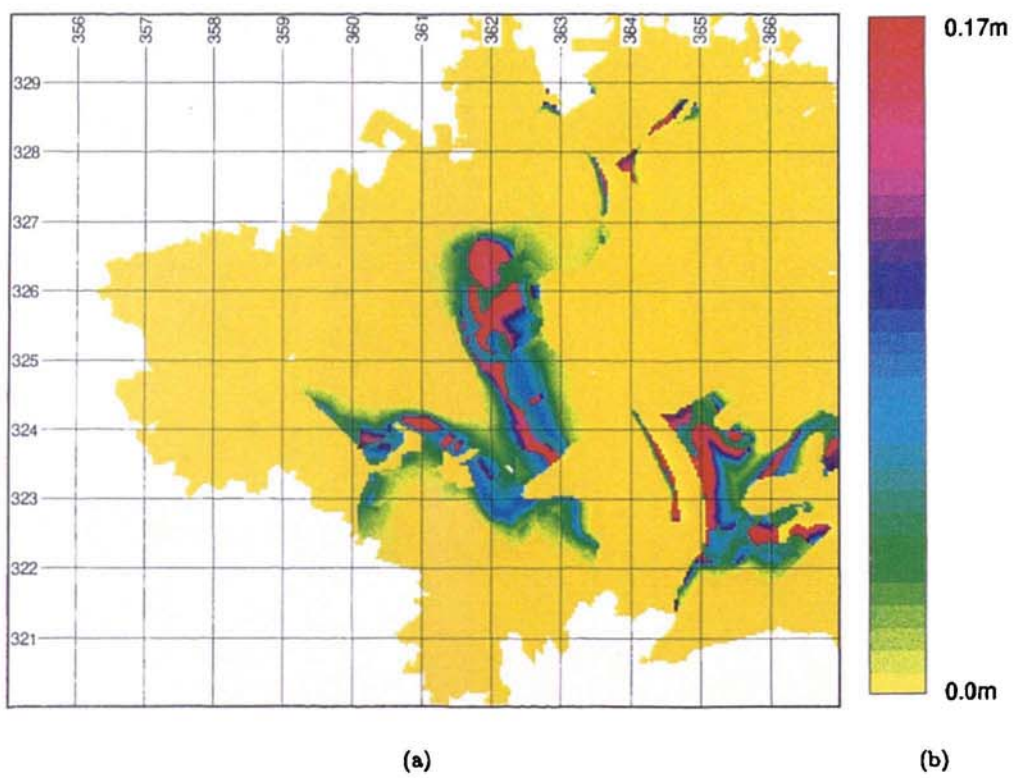


Figure 4.6: A raster showing calculated soil moisture losses.

```
area_of_pixel=(nsres() * ewres())/10000, \
lpp=(duration_of_moisture_loss * moisture_loss7 * 1000 - 104)8/area_of_pixel),0)
```

The distribution of losses is shown in Figure 4.7

In order to get a figure of loss for the whole region it is simply necessary to sum the loss in each pixel multiplied by the area of the pixel (ha). Surprisingly GRASS lacks the capability to do this directly but with an *awk* script this can be easily achieved.

```
r.stats -ln yield_loss >temp
awk 'BEGIN {total=0} {total=total+$1} END{print total * 0.16}' temp
```

First the GRASS program *r.stats* is run. The flags 'l' and 'n' cause the program to write a value for each pixel missing out null values. This is then redirected into a file "temp". *Awk* is a standard UNIX tool used to manipulate files (see Section 3.4.3). In this case it simply reads "temp" and totals all the numbers. The total yield loss of cereal is found to be 434 tonne, comparable with Hedges's (1989) figure of 428 tonne.

Hedges went on to establish a cash price for the loss of crops. This is not repeated here as prices have almost certainly changed, and it is merely a question of multiplying the yield loss by the price per tonne.

It should be noted that this figure for cereal loss is based upon Skinner's (1979) scenario. This scenario is based on a computer model of pumping all the proposed boreholes; Tern 1, 2 and 3 groups. Currently only Tern 1 group is operational, Tern 2 and 3 have yet to be implemented. Furthermore there is considerable uncertainty in the data, this is discussed in the next section and the model ignores many other factors such as irrigation or the crops roots following the watertable down etc.

4.6 Discussion

The purposes of repeating Hedges's (1989) analysis are; firstly, as stated in Section 2.3, one of the aims of this project is to use a general GIS in place of specifically written software, and to see how practical this is; and secondly, to be able to critically examine the method and techniques proposed by Hedges with a view to improving them. It was not the purpose to provide results to form the basis of any decision making. During his research Hedges performed a very large literature search to provide the basis for the values, such as the crop rooting depths, that he used in his analysis. Though several years have passed since Hedges's (1989) work this search was not repeated to add to the database of values. This is because firstly it takes considerable time and because, to state again, the purpose of this project, and this chapter specifically, is to see if Hedges's (1989) work is practical on a general purpose GIS, to critically examine the technique and not to produce results for any practical decision making.

⁷ This raster is generated in Section 4.4 above, but using the appropriate root depth.

⁸ This is the yield loss term, Equation 2.15.

4.6.1 Practicality

It was found that Hedges's (1989) work can be performed on a general purpose GIS. In GRASS, though some programming is required this is mostly the specification of queries to the *r.mapcalc* facility. In some cases low level programming has been required, for example the addition of functions to *r.mapcalc*, but *r.mapcalc* has been written so that such additions are possible.

The advantages of using a GIS are clear. Firstly the GIS comes with a great number of functions for manipulating spatial data. In the analysis above, frequent use was made of *r.surf.contour* program, a very sophisticated approach to handling contours which attempts to mimic the way contours would be manually interpolated. To write this and all the other programs used from scratch would be to rewrite the GIS a waste of time.

When Hedges wrote his system, GIS were not freely available. His system can be viewed as a GIS but one with a very restricted subset of the functions normally found in a GIS, and one that is very inflexible should other tasks be required of it.

4.6.2 Evaluation of Hedges's (1989) technique

The obvious question to ask about Hedges's (1989) technique is how good is it? There is an important corollary to that question and that is what purpose is it being put to? This was a point made in Section 3.1 the worth of a model can only be judged in the context of the use it is going to be put to. Hedges makes the point several times that his system is being developed as "a planning tool".

Hedges's (1989) technique is based on a conceptual model of the way the soil moisture interacts with the environment. As with any model there are three aspects to how good the model is. Firstly; does the model describe the significant aspects of the system under investigation? Given that it does, then secondly; how well does the model simulate the system of interest in the context of the purpose? And thirdly; how practical is the model to use in terms of the resources required from the user?

The accuracy of the model must be questioned. Hedges has encoded his model into a complete system, in order to get all the required parameters he has had to make numerous assumptions; about the depth of crop roots, the effects of clay drift, the relationships between yield and soil moisture, the relationships between particle size and various soil properties, the homogeneity of soil properties.

The way in which the system is used will be significant in deciding whether it is fit for the task. Hedges designed the system as a tool for planners. As such it would be inappropriate to run the system just once and accept the output result as the only possible outcome of the development or even the most likely outcome. One way in which the system might be used is the way that Hedges used it, what might be called the "worst case scenario". Following this principle Hedges chose to use the maximum rooting depth he found in the literature rather than the average. Figure 4.8

shows a sequence of rasters showing the spatial yield loss using different rooting depths. As expected as a larger rooting depth is assumed so the yield loss increases for a given drawdown pattern. Figures 4.8.a, b & c are just three possible depths. By embedding the query within a shell script (see Section 3.4.3) values can be generated for many different depths and plotted as a graph. Figure 4.8.d shows that over this range using Hedges's (1989) model the yield loss is approximately proportional to the assumed depth and some indication of the sensitivity of the depth parameter is given.

An alternative way of using the model and improving its fitness is to establish a range of outputs by varying the inputs. All of the inputs, whether soil properties, rooting depths, harvest times etc. can be expected to vary spatially. Without extensive surveying it would be impossible to capture this spatial variability. For the system that is being used to evaluate the effects of future development (as Hedges intended) then it is not possible at all. Hedges divided his landscape into units, e.g. crop types and soil series, and each unit was given a constant value. In effect he made the assumption that the spatial variability was adequately captured by the units he chose. An alternative is to attempt to capture the variability in a stochastic way. This subject is discussed further in Section 6.7.2.

Hedges's (1989) technique is built up in layers of reasoning. So even if the yield calculated is still considered too uncertain it is possible to "back track". That is to say if the yield is unsatisfactory, the distribution of soil moisture loss would still provide value insights for a planner. If the soil moisture loss distribution is not satisfactory then the areas of actual or potential effects could be used. So it is possible to conclude that as a tool for planners the system as it stands is adequate provided it is used sensibly.

Apart from deliberately adopting a worst case scenario and delineating potentially and actually vulnerable areas, Hedges did not perform much of this kind of analysis. This can be attributed to the constraints of the hardware used. For example, because of memory constraints it was only possible to run EIACOST on four 1 km squares at a time.

In terms of practicality the technique scores highly, as many of the features that bring its accuracy into question make it easy to use and understand. Compared with a distributed process model the data requirements are low once the empirical relationships have been established. However, these relationships represents a weakness in Hedges's (1989) approach. His relationships and values were derived from all sorts of sources: the crop relationships were derived from experiments in a different area (though on similar soil); the rooting depths from several sources etc., there is no satisfactory way of checking these within the Tern region without considerable expense.

The conceptual model itself is easy to understand; importantly this means its weaknesses are obvious to the user, and are not hidden behind a sophisticated mathematical solution.

4.6.3 Outline of the next stage

In order to tackle some of the problems and issues discussed above, the current project was conceived. Since Hedges's (1989) study there has been considerable development in computers and computing. GIS now exist that, as has been shown, can perform the tasks that Hedges was obliged to write a special system to do. However, computers are now also powerful enough to handle modelling complex environmental systems that Hedges could only handle using grossly simplifying assumptions. The next step is to use this new potential and to examine how a system, developed from a standard GIS and from environmental process models, might be constructed.

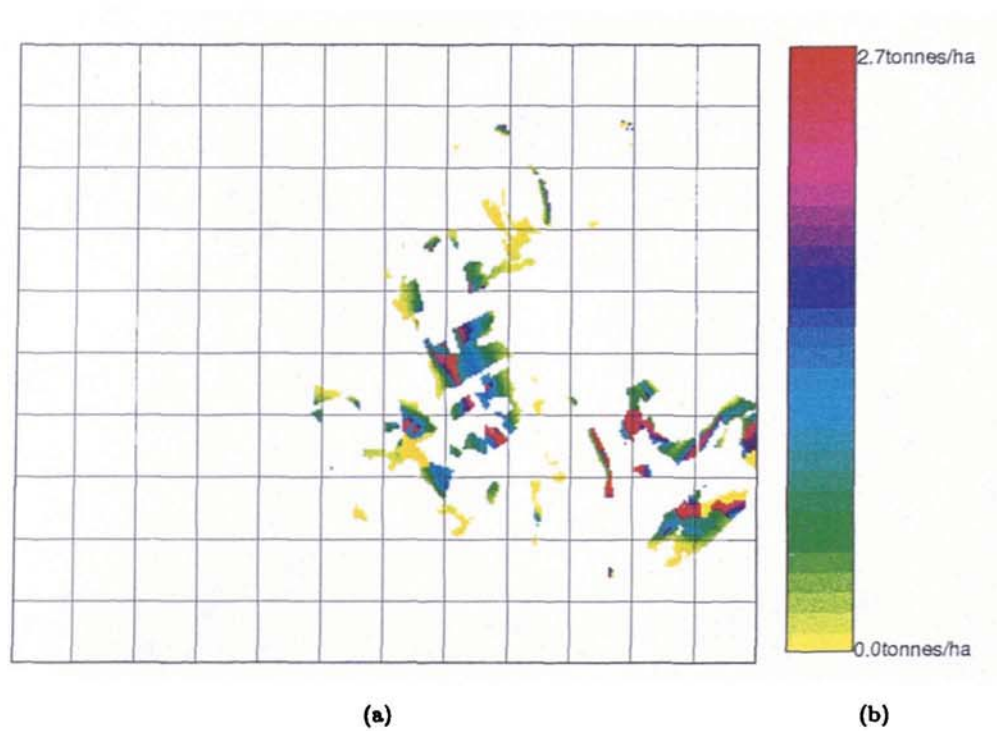


Figure 4.7: A raster showing calculated cereal yield losses (tonnes/hectare).

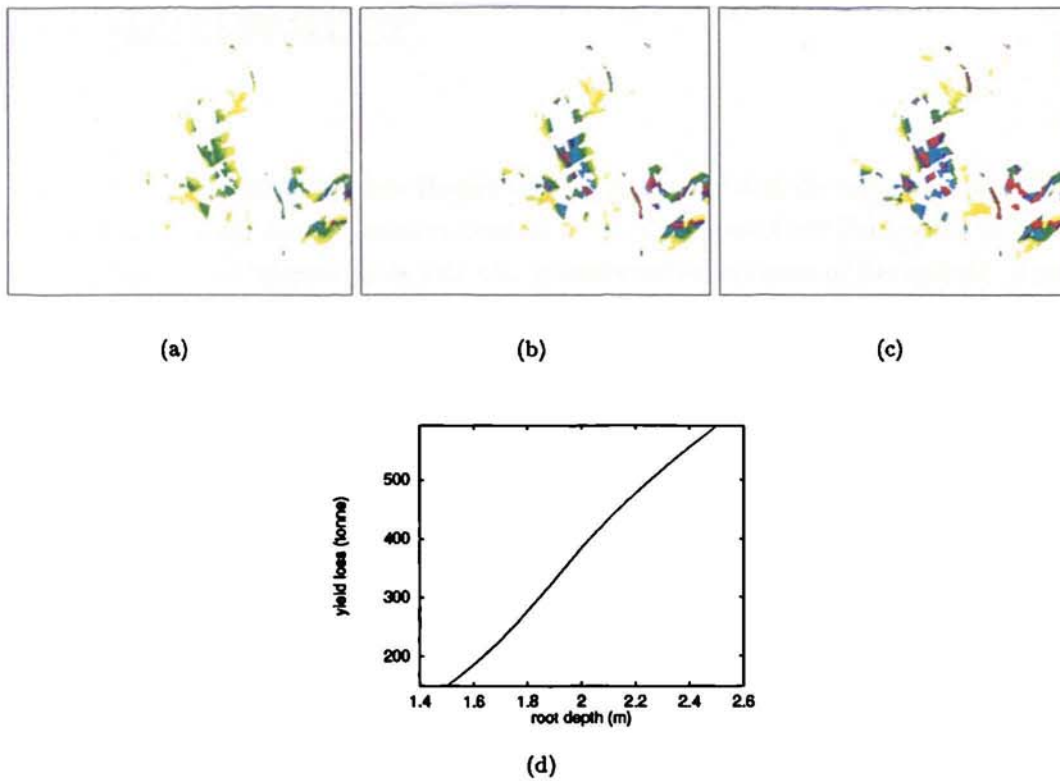


Figure 4.8: The effects of varying the root depth parameter on the calculated cereal yield loss, (a) yield loss per hectare with a root depth of $1.5m$, (b) root depth of $2m$, (c) root depth of $2.5m$ and (d) total yield loss against cereal root depth.

Chapter 5

Groundwater

In Section 2.2.1 it was discussed how Hedges's (1989) contended that the movement of the groundwater table could affect the soil moisture regime in the Tern area of the Shropshire Groundwater Scheme, therefore this chapter deals with the groundwater component of the system. Clearly, in this research project, it is the position of the watertable through time that is the single most important result that is required from the groundwater model, as it is intended that this parameter be input into a soil moisture model to calculate the resultant soil moisture profile. Section 5.1 briefly reviews the theory of groundwater and groundwater modelling. Section 5.2 looks at groundwater modelling exercises documented in the literature, with Section 5.3 reviewing the use of GIS for groundwater projects. Section 5.4 describes the linkage between the groundwater model and GRASS. Section 5.5 characterises the groundwater system in the Tern region of Shropshire. Section 5.6 describes a statistical investigation of the groundwater system, and Section 5.8 describes some of the modelling exercises that were performed with the GIS-model link.

5.1 Groundwater and groundwater modelling theory

The science of groundwater is mature, and for an in-depth discussion the reader is referred to the many good books on the subject such as DeWeist (1968) or Bouwer (1978). Current research is focused upon; studying real complex aquifer systems using new techniques involving tracers, geophysics and packer pump testing; chemical interactions within aquifers including man-made chemicals; and long term trends and socio-economic aspects of aquifers. Similarly groundwater modelling is also well established (Rushton and Redshaw, 1979), with current research concentrating on complicating factors e.g.; contaminant transport, multi-phase flow, ocean-aquifer interaction, vertical characterisation of the aquifer (Price, Morris and Robertson, 1982) and fracture flow. This project uses well established groundwater science and modelling techniques, consequently only the briefest review will be attempted here.

A working definition of groundwater is: the water that has seeped into the ground, whether



Figure 5.1: Typical features of a groundwater system.

into soil, rock or unconsolidated drift such that the media is fully saturated and the pressure greater than atmospheric. Figure 5.1 illustrates some of the terminology employed with respect to groundwater and many typical situations that can be encountered.

Groundwater systems are fields of potential energy and flow moves from regions of high potential to regions of low potential along the line of the steepest gradient. This is expressed by Darcy's law:

$$Q = KA \frac{dh}{dx} \quad (5.1)$$

where:

- Q — water flow rate between two points (L^3T^{-1});
- $\frac{dh}{dx}$ — instantaneous hydraulic gradient;
- K — hydraulic conductivity (LT^{-1}), dependent upon permeability of the aquifer material and fluid properties;
- A — cross-sectional area (L^2).

Ignoring physical models, groundwater models come in two varieties. Firstly, the simple mass balance models used to estimate resources over long periods. Skinner (1979) used such a model to estimate the potential yield from the Shropshire Groundwater Scheme in his evidence to the

Public Inquiry. The other more complex models use finite element or finite difference methods to model flow and solute movement in the aquifer. These models use a one, two or three dimensional formulation of Darcy's law and assumptions of continuity to simulate flow:

$$K_x \frac{\partial^2 h}{\partial x^2} + K_y \frac{\partial^2 h}{\partial y^2} + K_z \frac{\partial^2 h}{\partial z^2} = S_s \frac{\partial h}{\partial t} \quad (5.2)$$

where:

- $K_{x,y,z}$ hydraulic conductivity in the x, y , or z directions (LT^{-1});
- S_s specific storage, the amount of water released from unit volume of aquifer with unit drop in hydraulic head (L^{-1}).

The aquifer region is divided into elements, and for each element the equation (5.2) is formulated in a finite difference form. Sink or source terms are added to account for loss and gains to and from boreholes, rivers, drains, recharge and evapotranspiration. The finite difference expression, expressed for each active unit, forms a set of simultaneous equations that are solved, almost inevitably, with an iterative approach. Boundary conditions are expressed by elements designated as *no-flow* or *fixed-head*. Modflow, McDonald and Harbaugh (1988), is an example of such a model and was selected for this study as described in Section 5.4.

A frequent assumption made in modelling is that the movement in the groundwater system can be adequately simulated by considering the two horizontal dimensions. Historically vertical variations in conductivity and storage have been integrated into the parameters of transmissivity and storage coefficient, for the mundane reason that vertical variations are hard to establish, and the act of performing a pump test results in this integration. Transmissivity is:

$$T = Kw \quad (5.3)$$

where:

- T transmissivity (L^2T^{-1});
- K average hydraulic conductivity of saturated thickness of aquifer (LT^{-1});
- w saturated thickness of the aquifer (L).

Storage changes between confined and unconfined conditions, so it is usual to talk about confined or unconfined storage coefficient. Storage coefficients are given by:

$$S = S_s w \quad (5.4)$$

where:

- S storage coefficient (dimensionless);
- S_s — average specific storage of aquifer (L^{-1}), the volume of water released from a unit volume of aquifer by a unit reduction in hydraulic head;

w saturated thickness of the aquifer (L).

Several authors have identified pitfalls that might be encountered when using numerical groundwater models. Reddi (1990) listed these as:

- “overkilling”, or undue sophistication in selection of models;
- misconceptualisation of the groundwater system;
- inadequacy of parameters;
- misinterpretation of model results.

5.2 Modelling experiences

The Triassic Sandstone, of which the Bunter Sandstone is a part, is a layer that forms aquifers of economic importance in several parts of the country, consequently it has been the subject of considerable research and hydrogeological modelling. These studies provide a valuable reference source when constructing a model, since frequently assumptions and simplifications have to be made and the literature is a guide to what may be acceptable and what is not. This section gives a brief overview of some of the research and modelling of relevance to this project, and will be referred to in later sections.

5.2.1 Birmingham aquifer

Knipe et al. (1993) undertook a study for the Construction Industry Research and Information Association (CIRIA) concerning the problem of groundwater rise in the Birmingham aquifer. Their study involved modelling the aquifer, which like the Shropshire aquifer is of Triassic sandstones.

In several ways their modelling effort is similar to that presented here. They too used Modflow, they have had to make assumptions about the aquifer due to paucity of data and they used the model to analyse different scenarios. The model was applied to an area approximately 180km^2 in Birmingham, to simulate a period of approximately 150 years. The model was a single layer 40 by 48 grid with a resolution ranging from 250 to 1000 metres.

5.2.2 Nottingham Sherwood aquifer

In response to a perceived long term decline in groundwater levels in the Nottinghamshire aquifer and the possible threat this posed to surface water conditions, the National Rivers Authority (NRA) began a study into the groundwater resources of the aquifer. The University of Birmingham was commissioned to develop a numerical model of the aquifer (Rushton and Bishop, 1993). The researchers at Birmingham University listed the aims of their modelling exercise as:

- to conceptualise the major flow mechanisms operating within the aquifer;
- construct numerically and hydraulically accurate mathematical models of the Nottinghamshire aquifer;

- to represent the historical flow balance of the aquifer;
- to evaluate an historical flow balance for the aquifer;
- to demonstrate the use of the model in the evaluation of the aquifer response under different management scenarios.

The period of the study was from August 1990 to September 1993 and, as well as using secondary sources of information, new investigations were also required. This represents a considerable amount of work.

Before any numerical modelling could be performed the researchers first had to develop a conceptual model of the aquifer behaviour. This involved identifying boundaries of the system, estimating initial aquifer parameters (transmissivity and storage coefficients), estimating recharge, river interaction, and aquifer outflows. They tested the validity of these factors by completing a mass balance analysis of the system.

Rushton and Bishop (1993) stated:

“ The absolute transmissivity values derived from pumping test analyses in the unconfined have been considered to provide only an approximate guide to the effective values required for the mathematical model. ”

So rather than define absolute transmissivity and storage coefficients, Rushton and Bishop identified trends. To this end they assembled all the available pumping test data and re-examined them. From this they delineated zones and assigned values of storage and transmissivity to these zones. Because of paucity of data, they made the assumption that values did not change over time with saturated thickness because they estimated annual oscillations only represent 1–2% of the thickness. They also assumed conditions were isotropic, i.e. transmissivity does not vary with direction. Unconfined storage coefficient was assumed to be constant.

Rushton and Bishop developed a distributed regional recharge model. They based it on a soil moisture balance approach, and assume that recharge may only occur when soil deficit is zero. Precipitation was measured by a network of 21 rain gauges, evapotranspiration was taken from Meteorological Office Rainfall and Evaporation Calculation System (MORECS).

Surface water was identified as being an integral component of the groundwater system and developed a model of river-aquifer interaction (Figure 5.2):

$$\begin{aligned} Q_r &= R(h - z_r) & \text{for } h > z_b \\ Q_r &= R(z_b - z_r) & \text{for } h \leq z_b \end{aligned} \quad (5.5)$$

where:

Q_r — the flow from aquifer to the river (L^3T^{-1});

R — the river coefficient (L^2T^{-1});



Figure 5.2: Rushton and Bishop's (1993) model of aquifer river interaction.

- h the groundwater head (L);
- z_r the river surface elevation (L);
- z_b the river bed elevation (L), which is assumed to be $1.0m$ below the river surface.

Rushton and Bishop give a typical value of $1000m^3d^{-1}$ for river coefficient for larger rivers of the region.

The numerical model itself was a two dimensional finite difference model, using a spatial discretisation of grid squares of $1km$. It was validated by comparing well hydrographs with modelled hydrographs, by examining the cumulative volumes to and from rivers and by examining the model aquifer mass balance. One of the conclusions reached was that:

“ the model can provide reliable predictions provided that the conditions remain within those for which the model has been verified; more radical changes in aquifer conditions may require modifications to the modelling parameters and possibly techniques. ”

The study of Rushton and Bishop (1993) shows a lot of similarities to the groundwater modelling presented in this thesis. It is in similar geology, has a mix of confined and unconfined conditions, there are river interactions and groundwater abstraction and a similar level of available data. Unfortunately this study only came to light towards the end of this research on the Shropshire Scheme. Differences between the two studies and indications of how the Nottinghamshire study was successful whilst the author's own study was less successful, are discussed in Section 5.9.



Figure 5.3: SRA (1974) groundwater model boundaries. The grid shows the size of grid used by the model.

5.2.3 Previous modelling of the Shropshire aquifer

The aquifer of interest to the study has also been the subject of earlier modelling. Of course Skinner (1979) presented the results of a mass balance model when he gave evidence regarding the likely yield of the SGS. In addition the Tern area itself has been the subject of a distributed numerical model, which is documented in SRA (1974). The model grid is shown in Figure 5.3 it was a two dimensional model with a grid resolution of 1km , and the coarseness of the grid necessitating a simplification of the boundaries.

Though pumping test results were available, these were considered unreliable, the method of determining spatial distribution of transmissivity parameters adopted was to use a steady state approach. In this approach recharge is applied to the model and transmissivity is adjusted until a satisfactory steady state surface is obtained.

One of the most important pieces of information to be gleaned from this work is the boundaries

that the researchers choose, Figure 5.3, as this will help in specifying the boundaries in the author's own model. The most obvious feature is how boundaries have had to be moved and simplified to accommodate the choice of resolution (1km). However, they do represent a very good starting point for specifying boundaries in a new model of the region.

5.2.4 Evaluation

Groundwater numerical modelling is performed around the world from the very large or site scale models (Ashley, 1994; Dixon, Bradford, Cooper, Reeve and Tucker, 1989) through the regional scale models (Allewijn, 1988; Rippon and Wyness, 1994) to the very small or national scale models (Sophocleous, 1992; Vermulst et al., 1996). The simplifying assumptions that have been made in these modelling efforts vary from those with very simple assumptions to those where very detailed vertical and horizontal data is available (Turner, Ervin and Downey, 1991). Only a few have been described above; those that are similar in geology, scale and intention to the modelling attempted in this project. The reasons for modelling are legion; preventing salinity build up in soil (Salama, Laslett and Farrington, 1993), predicting the effect of groundwater changes on buildings (Knipe et al., 1993), pollutant movement (Michl, 1996) and the possible mobilisation of radioactive waste (Turner et al., 1991). Since groundwater modelling is such a widespread activity, there are many guidelines and recommendations for effective use of groundwater models (e.g. see EPA, 1992; Grondin, Gannett, van der Heijde and Patt, 1990). However, nowhere in the literature was an example found where the output required was a watertable level so accurately specified through space and time as in this project where precision to a few centimetres would not be too great although on examining the literature this is optimistic.

5.3 GIS and groundwater investigation

Groundwater processes vary spatially, and consequently there are an increasing number of examples from around the world of investigations that have exploited the spatial analysis capabilities of GIS for mapping and quantifying groundwater processes. The following are a selection of such examples drawn from recent literature.

D'Agnese, Faunt and Turner (1996) used remote sensing and GIS technology to estimate the spatial variation in recharge and discharge for the Death Valley groundwater system. Satellite images were used to derive a map of vegetation, which was combined with other data and existing evapotranspiration relationships to estimate discharge volumes from identified discharge areas. The spatial recharge estimates were derived using, essentially, a cartographic modelling technique (Section 3.3.3). Maps describing elevation, aspect, permeability and vegetation were combined to produce a recharge potential map which in turn was combined with actual precipitation rates to produce a recharge rate map.

Engels, Navular, Cooper and Hahn (1996) were concerned with estimating groundwater vulnerability to non-point pollution, such as nitrates and pesticides. They compared three models of groundwater vulnerability combined with a GIS. One model they used was DRASTIC. The DRASTIC acronym is made up of the elements **D**epth to water table, **R**echarge, **A**quifer media, **S**oil media, **T**opography, **I**mpact of the vadose zone and **C**onductivity. Each factor is assigned a weighting based on its significance and in any given area each factor will have a measurable rating. The DRASTIC index is the total of the product of weight and value for each factor. The higher the index the greater the potential vulnerability to pollution. The DRASTIC model is a cartographic model and easily implementable with a map algebra language.

Fels and Matson (1996) and Matson and Fels (1996) used a GIS to map watertable depth at a regional scale. They used the GIS to divide the landscape into landscape types; e.g. mountains, hills, flood plains and wetlands, and an automatic techniques to subdivide these areas into sub-regions, on the basis of landscape position; i.e. whether on the bottom the sides or the top of a valley. These were then assigned to watertable depth classes. Again this is another cartographic modelling technique (the index they derived is adopted as an explanatory variable in Section 5.6.3 and illustrated in Figure 5.37).

5.3.1 GIS and numerical groundwater modelling

Recently there has been a surge in activity relating to the use of GIS as part of groundwater numerical modelling exercises. This work originated from the perception that GIS would be a suitable technology to handle the estimation of input parameters. Subsequently links have been developed between GIS and groundwater models along the lines discussed in Section 3.4.2.

Ross, Fielland and Tara (1990), Ross, Schwartz and Tara (1991) and Ross et al. (1991) describe one of the earliest uses of GIS and numerical groundwater modelling. They produced a modelling system by linking a commercial GIS, with public domain surface and subsurface models and specially written code. The system was intended to support phosphate mining reclamation design.

Harris et al. (1993) developed a similar system for analysing flow in the San Gabriel Groundwater Basin, USA. Both Ross and Tara (1993) and Harris et al. (1993) are examples of the loosely linked approach described in Section 3.4.2.

Kamps and Oltshoorn (1996) described an approach where the GIS was used for more than data handling. They have linked Modflow and Arc-Info and are using the GIS to embed fine resolution local models within coarse regional scale models. They are thus increasing the capabilities of the GIS linked Modflow beyond those of the stand-alone model.

The linkage of GIS and groundwater investigation and modelling has been one of the major driving forces in a recent evolution of GIS. Several authors have described the development of 3-dimensional GIS, and a few have taken the step of categorising them as Geo-Scientific Information Systems (GSIS).

3-dimensional GIS, along with temporal GIS, represent the cutting edge of GIS research,

GSIS have not yet matured into using a standard set of data models and structures, as their 2-dimensional cousins have, and the commercial supply of such systems is a very small and specialised sector. Examples of these systems which have been applied to groundwater numerical modelling include Fisher (1993) and Turner et al. (1991).

5.4 Modflow and GRASS

Having identified the groundwater system as an important component of the environmental system of interest, it was necessary to select and link a numerical groundwater flow model with the GIS GRASS. There was not the time to spend on an in-depth investigation of all the available models, against all the criteria discussed in Section 3.2.2, and it was thus decided to adopt the most well used model, as this satisfied all the required criteria:

- simulation of flow and provision of spatial distributions of groundwater surface and draw-down;
- compatibility with a GIS;
- ability to handle very large datasets;
- ability to model borehole abstraction, confined and unconfined situations, river interaction etc.;
- preferably be public domain or inexpensive;
- desirable that it should be easily modified, i.e. source code should be available.

The model adopted was Modflow (McDonald and Harbaugh, 1988). This is a public domain 3D numerical finite difference groundwater flow model developed by the US Geological Survey. It is almost certainly the most widely used groundwater model (it was used in several of the examples above, Section 5.3.1).

McDonald and Harbaugh (1988) derived Equation 5.6 to describe the change in storage of a variable head cell in a time-step.

$$Ss_{i,j,k}(\Delta r_j \Delta c_i \Delta v_k) \frac{\Delta h_{i,j,k}}{\Delta t} = CR_{i,j-\frac{1}{2},k}(h_{i,j-1,k} - h_{i,j,k}) + CR_{i,j+\frac{1}{2},k}(h_{i,j+1,k} - h_{i,j,k} + \\ CC_{i-\frac{1}{2},j,k}(h_{i-1,j,k} - h_{i,j,k}) + CR_{i+\frac{1}{2},j,k}(h_{i+1,j,k} - h_{i,j,k} + \\ CV_{i,j,k-\frac{1}{2}}(h_{i,j,k-1} - h_{i,j,k}) + CV_{i,j,k+\frac{1}{2}}(h_{i,j,k+1} - h_{i,j,k} + \\ P_{i,j,k}h_{i,j,k} + Q_{i,j,k} \quad (5.6)$$

where:

Ss — specific storage;

$\Delta h_{i,j,k}$ — change in head in cell i, j, k in the time-step t (L);

Δt — length of time-step t (T);

$\Delta r_j \Delta c_i \Delta v_k$ — width of the j th row, the i th column and the height of the k th layer (L);

$CR_{i,j-\frac{1}{2},k}$ — conductance in i th row and k th layer between columns j and $j - 1$ (LT^{-1});

$CR_{i,j+\frac{1}{2},k}$ — conductance in i th row and k th layer between columns j and $j + 1$ (LT^{-1});

$CC_{i-\frac{1}{2},j,k}$ — conductance in j th column and k th layer between rows i and $i - 1$ (LT^{-1});

$CC_{i+\frac{1}{2},j,k}$ — conductance in j th column and k th layer between rows i and $i + 1$ (LT^{-1});

$CV_{i,j,K-\frac{1}{2}}$ — conductance in i th row and j th column between layers k and $k - 1$ (LT^{-1});

$CV_{i,j,K+\frac{1}{2}}$ — conductance in i th row and j th column between layers k and $k + 1$ (LT^{-1});

$P_{i,j,k}$ — net conductance of external sources and sinks that are dependent on the cell head (L^2T^{-1});

$Q_{i,j,k}$ — net flow of external sources and sinks that are not dependent on the cell head (L^3T^{-1}).

Modflow is constructed in a modular fashion, this is so that a user constructing a model can use only those elements of the groundwater system of relevance to their problem, and so that in future extensions to the model can be added easily. The principal modules that have to be included in any model are:

- the *basic* module, which controls the geometry of the system; the number of rows, columns and layers, the codes assigned to elements in the model (variable head, fixed head or no-flow) and the number and length of stress periods and time-steps;
- the *block centred flow* module. Which controls the assignment of aquifer properties, (storage, transmissivity and vertical conductivity) to elements;

and either

- the *strongly implicit procedure* (SIP) module;

or

- the *successive over-relaxation procedure* module;

both of these last two modules are for solving the simultaneous finite difference equation. Other modules which can be added are:

- the *rivers* module, for simulating aquifer-river interaction;
- the *drainage* module, for simulating the effects of drainage systems such as tile drainage;
- the *well* module, for simulating abstraction from or injection to boreholes or any other kind of point source or sink;
- the *recharge* module, for simulating recharge to the aquifer;
- the *evapotranspiration* module, for simulating loss to the atmosphere from the aquifer.



Illustration removed for copyright restrictions

Figure 5.4: Conceptual model of the river bed used by McDonald and Harbaugh (1988) in Modflow.

5.4.1 Rivers package

The *rivers* module enables Modflow to simulate the interaction between the aquifer and a surface water feature. In reality this kind of situation is complex and often ill-defined, and Modflow makes a number of assumptions in order to make the problem tractable. The assumptions that are made strongly resemble those of Rushton and Bishop (1993) — see (Section 5.2.2).

McDonald and Harbaugh (1988) use a simple model of the river bed, as illustrated in Figure 5.4. As well as being dependent on the river bed material, conductance depends on the area of the river bed. For each cell it is necessary to work out the length of river passing through it in order to calculate the river conductance.

$$C_{riv} = K_{riv}WL \quad (5.7)$$

where:

C_{riv} — conductance of the river bed (L^3T^{-1});

K_{riv} — conductivity of river bed (LT^{-1});

W — width of river reach (L);

L — length of river reach (L).

As the relative head between the river and the cell it is in changes, so does the flow rate between them.

$$\begin{aligned} Q_{riv} &= C_{riv}(H_{riv} - h_{i,j,k}) & \text{for } h_{i,j,k} > R_{bot} \\ Q_{riv} &= C_{riv}(H_{riv} - R_{bot}) & \text{for } h_{i,j,k} \leq R_{bot} \end{aligned} \quad (5.8)$$

where:

Q_{riv} net flow in a cell from river to cell (L^3T^{-1});

H_{riv} stage of the river (L);

$h_{i,j,k}$ hydraulic head of cell i, j, k (L).

As with other packages the rivers packages operates on a cell by cell basis. For each cell affected by seepage, the flow to or from the aquifer is calculated and then added to the groundwater flow equation (Equation 5.6). McDonald and Harbaugh (1988) make the point:

“ ...it should be recognised that formulation of a single conductance term to account for a three-dimensional flow process is inherently an empirical exercise, and that an adjustment in calibration is almost always required. ”

5.4.2 Using Modflow

Modflow is in the public domain and the source code is available to download using the Internet from various sites around the world. The first task was to compile the source code and make those changes that were necessary to allow it run on the hardware used Sun IPX's , and Sun Sparcstation 4's and 5's with, at the time, SunOS 4.3 operating system, under Open Windows . It is well written in FORTRAN — it complies strictly with the FORTRAN 77 standard — so minimal changes were necessary, (in fact just one line, a new value for a constant and a few comments were added to the *main.f* file). The model was tested with data provided with the documentation. and apparently worked well. It has since been recompiled several times as the operating system on the UNIX network was upgraded during the course of the project, and no further changes to the source code proved necessary.

Modflow uses a data file interface. That is to say that all communication from the user to the model is via the input data files constructed by the user before the model is called and run. These files provide everything that Modflow needs to know. There is a file for each module used. plus a basic information file and an output file that tells Modflow how to construct its output files. When the input files have been constructed Modflow can be run and if the files have been properly constructed then Modflow will generate some output files.

5.4.3 The Modflow – GRASS link

The Modflow – GRASS link was constructed according to the strategy outlined in Section 3.5.1. The link as it stands represents the first prototype. Only those modules of Modflow relevant to the project in hand have been linked, these are:

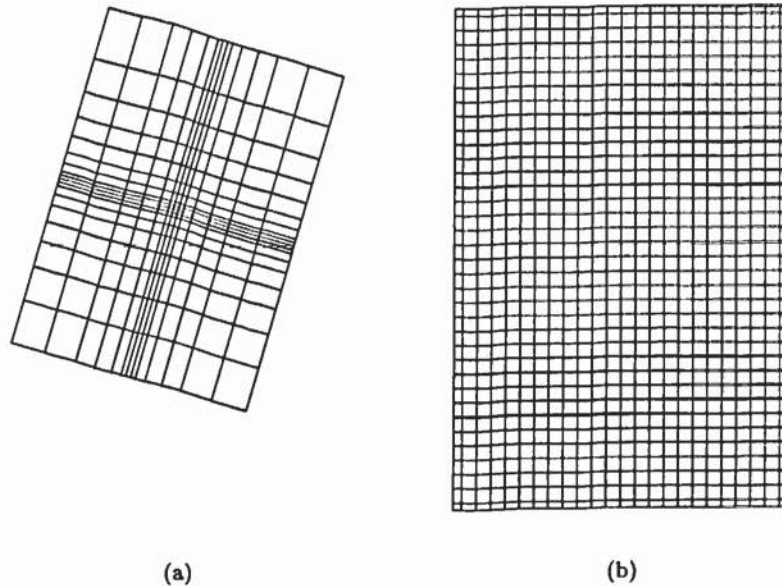


Figure 5.5: The difference between (a) Modflow and (b) GRASS grid models, The GRASS model is the classic raster model aligned north-south, the Modflow grid has variable resolution.

- basic Module;
- block centred flow module;
- recharge;
- evapotranspiration;
- rivers;
- wells;
- strongly implicit procedure.

Several programs were written which together effect the linkage.

- *mod.edit* — the main program for editing the dataset;
- *mod.write* — writes a dataset into the form of the Modflow input files;
- *mod.run* — sets up and runs Modflow on the files generated by *mod.write*;
- *mod.del* — deletes a Modflow dataset;
- *mod.rename* — renames a Modflow dataset;

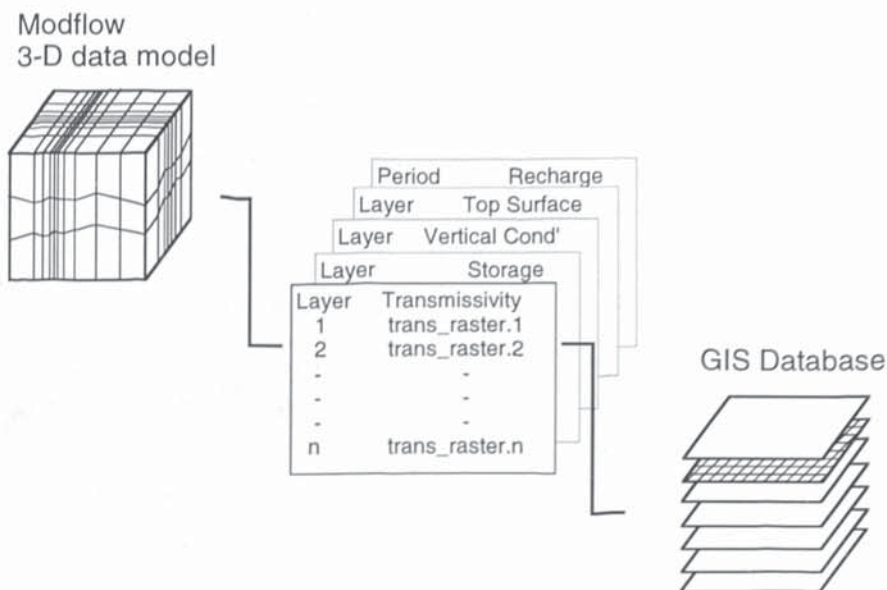


Figure 5.6: The link between Modflow and GRASS is essentially formed of lists that identify a GRASS raster with a layer or stress period property in the Modflow model.

- *mod.copy* — copies a Modflow dataset;
- *mod.xtrct* — extracts data from the Modflow output files and writes them into the GRASS database;
- *mod.d.grid* displays an irregular grid in the current GRASS monitor;
- *mod.cre* non-interactively creates an empty Modflow dataset;
- *mod.region* — changes current GRASS region to a Modflow dataset region;
- *mod.rem* performs a bulk remove of rasters from the GRASS database;
- *mod.ts* extracts values from a family of rasters from a point to form a time.

The main issue is reconciling the data structures between GRASS and Modflow, in particular between the two dimensional GRASS raster and the 3D Modflow grid. This is illustrated in Figure 5.6. Modflow and the GRASS database are linked quite simply by a set of lists that describes which raster in the GIS corresponds to each attribute of each layer and each period in the model. The user interface can be as simple as a text editor to edit these lists.

Most of the data modules within Modflow accept data as in an irregular grid structure, Figure 5.5.a. The dimensions of the grid are specified in the basic input file, and this acts as the

```

Enter further details for layer:1

one transmissivity   if 'y' what   if 'n' enter raster to valid
for all rows y/n?   is value?   read values from
      n             0.0000000000   layer1_trans_____ y

one vertical         if 'y' what   if 'n' enter raster to valid
conductivity 2/n?   is value?   read values from
      y             0.5000000000   _____ y

one secondary        if 'y' what   if 'n' enter raster to valid
storage y/n?        is storage   read values from
      y             0.1000000000   _____ y

one top level for    if 'y' what   if 'n' enter raster to valid
whole layer y/n?    is height   read heights from
      n             0.0000000000   digital_elevation_model_____ y

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure 5.7: The Modflow-GRASS link *block centred flow* module interface. Here the rasters describing the properties of the top layer of the model are specified, i.e. transmissivity in the top layer is described by a raster called *layer1_trans*. Where data is scarce it may only be possible to give a single value to describe a whole layer, i.e. secondary storage (unconfined storage) is given a value of 0.1. The user indicates whether a raster or a single value is given by answering *y* or *n* to the first question. The interface is constructed with the GRASS visual ask (VASK) library (Appendix A.1.1).

horizontal discretisation of the flow domain, the vertical dimension is divided by specifying the number of layers in the model, the temporal dimension is divided by specifying a number of different length stress periods.

If a Modflow module is concerned with the three dimensional distribution of some parameter then this is presented to the module as a sequence of grids one for each layer, for example, transmissivity of each layer is presented as a sequence of grids to the block centred flow module. Where a module is concerned with the temporal dimension then again the data is specified as a sequence of grids, one for each stress period. Because many of the modules accept grids of data, a library of functions was developed to resample grids, write and read grids into the dataset stored in the GRASS data, to write grids into the model input files and interact with the user to specify connections between grids and rasters. Most of the interface (which is documented in Appendix B) is simply a way of enabling the user to link a GRASS raster to a Modflow layer. Figure 5.7 shows the interface to the block centred flow module, which uses these grids.

The link has been written to allow irregular grids which are aligned north-south to be used, it would have been quicker to implement if to Modflow had been restricted to using only regular grids, however, there are considerable speed and storage advantages to have an irregular grid. Modflow can account for anisotropic conditions. The Modflow grid should be aligned with the principal direction of hydraulic conductivity. No evidence was found of anisotropic conditions in

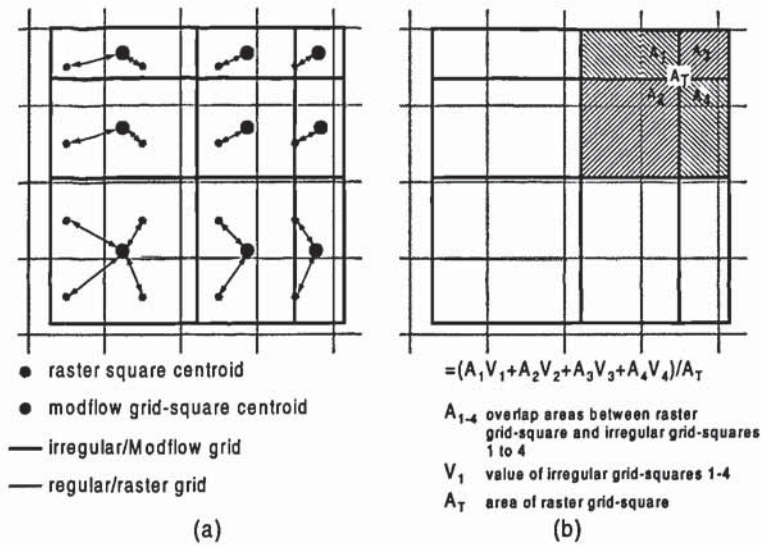


Figure 5.8: Methods of resampling between irregular grids and rasters; (a) nearest neighbour resampling, which is very quick and simple to implement and (b) area weighted average, this is less quick but maintains the total of the pre-resampled values.

the Shropshire aquifer in the literature, so an assumption of isotropy was made. This meant that the irregular grids could be constrained to being aligned north-south — considerably simplifying the resampling algorithms. *mod.write* uses a nearest neighbour or area weighted average method of resampling to convert between grids and rasters, Figure 5.8.a.

The rivers module and the wells module are written rather differently, they do not use the irregular grid structure, consequently different data structures are used to hold the data in the GRASS database. For the river data the vector structure is the most appropriate.

The weaknesses of GRASS's database was highlighted here. There was no simple way of storing all the attributes associated with each river length as in a GRASS vector structure each segment can only have one associated number and category stored with it. Consequently an *ad hoc* method was developed and the attributes of each river reach were stored as part of the Modflow dataset object in the GRASS database rather than as part of the vector object representing the river network.

When the river vector is to be converted to the Modflow input format, each individual vector within the vector file is retrieved in turn and the attributes for it are retrieved from the Modflow dataset, Figure 5.9.a. The vector is divided into its component two point segments, Figure 5.9.b. Each two point segment is divided up where it crosses a Modflow grid boundary, Figure 5.9.c. The length of each small arc is calculated, this is multiplied by the width and conductivity of the river

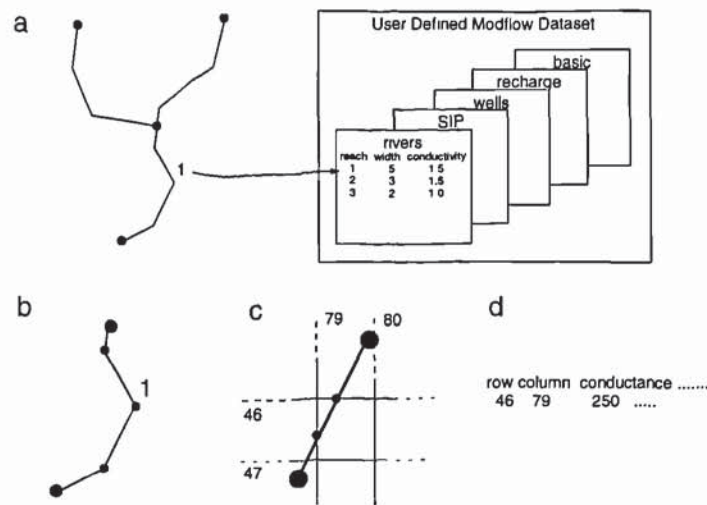


Figure 5.9: Steps in converting a GRASS vector structure to a Modflow river file; (a) each arc in the vector file is linked by the user to a set of attributes; (b) each two point segment of the arc is taken in turn; (c) the segment is divided into several shorter segments where it crosses the Modflow grid boundaries; and (d) the information about each short segment is written into the Modflow river file.

reach and written to the GRASS input file, Figure 5.9.b.

Each river segment extracted in this manner is considered to be rectangular, however GRASS treats vectors as being infinitely thin, i.e. width must be treated as a non-spatial attribute of the river reach. When a vector is expanded outwards by its width then, at the edges of Modflow grid squares, one finds fragments of river reaches overlapping adjacent grid squares (Figure 5.10). These are ignored and the whole area of the river reach is considered to be acting on the first grid square. This is acceptable when reaches are relatively thin when compared to grid square dimensions. At the resolution that Modflow is applied in Shropshire (typically 50-100m grid squares compared with typical width of the Tern of c.5m) this is valid. If Modflow were to be applied at a scale at which river widths became comparable to grid square sizes, for example if the model were applied to a single borehole next to a large lake, this would not be acceptable. A future prototype of the link could allow the user to chose whether to represent surface water features by vectors or rasters (see Section 8.3).

For the well module it was also considered appropriate to use a different GRASS structure to store the data, in this case the GRASS site structure was used. At almost every scale that Modflow could reasonably be applied to, a borehole or well will be a point. As with rivers the GRASS lacked sufficient database facilitates to manage the link between the spatial data (in this case the point) and the none spatial data, which would be a time series of pumping rates. So

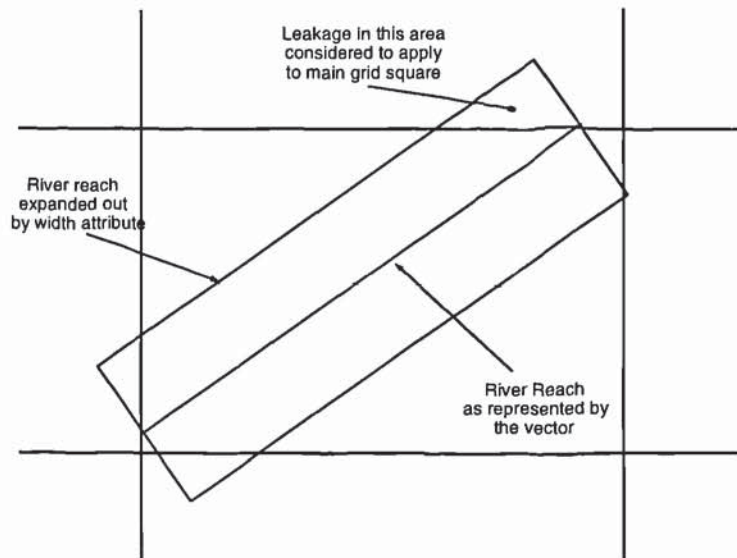


Figure 5.10: The river reach overlaid on a Modflow grid-square.

again an *ad hoc* solution was necessary.

The user selects a GRASS site file to represent the boreholes. For each site the user specifies a flow rate for each stress period (a flow rate which can be negative indicating injection into the aquifer), then the user specifies a fraction of this flow from each layer in the grid model. These values are stored within the dataset object in the GRASS database, Figure 5.11.

When the data is to be written to a Modflow input file, the site file is extracted from the GRASS database. For each site the corresponding record is extracted from the dataset object in the GRASS database. The link then calculates which grid square the borehole is in, works out the flow rate from each layer for each stress period and writes this to the input file.

When a dataset is complete it can be written into the Modflow input files. These can be large and are only generated when Modflow is to be run. After Modflow has finished the program *mod.xtrct* is used to write the generated groundwater heads and drawdowns back into the GRASS database. *mod.xtrct* can write the data to a raster or to a site file. It can extract all the output of just select layers and periods. As with *mod.write*, it can resample from an irregular grid using nearest neighbour or an area weighted average method (Figure 5.8).

5.5 Characterising the Tern groundwater regime

Before any numerical model can be constructed the system which is to be modelled must be adequately understood. This can be termed constructing a conceptual model¹. In the case of

¹ In the first sense of the term described in Section 3.2.

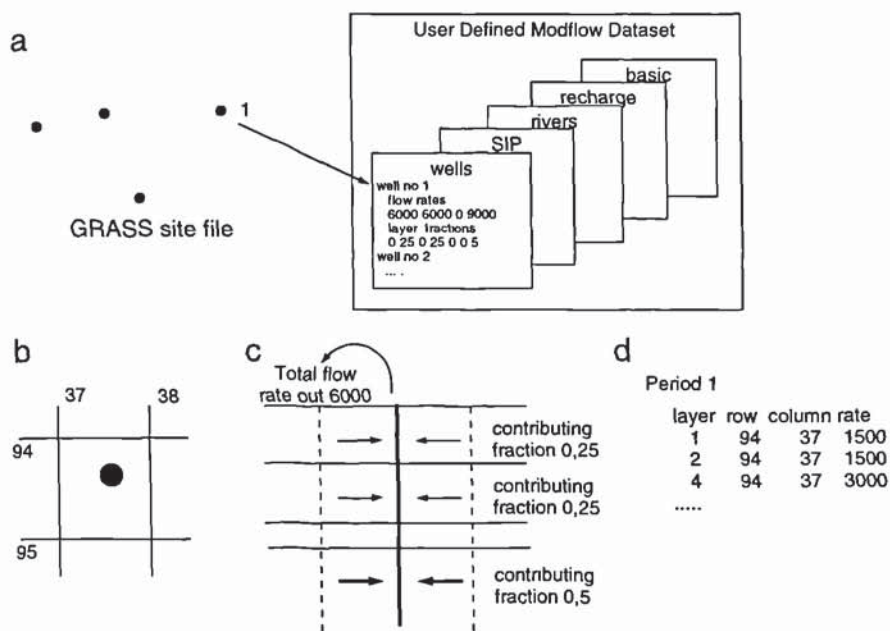


Figure 5.11: Steps in converting a GRASS site file to a Modflow well file. (a) Each site in a GRASS site file is linked with a set of attributes in the Modflow dataset; (b) when Modflow is to be run the site is located in a grid-square; (c) flow from the well is divided between the layers according to user defined flow fractions; and (d) the results are written into a Modflow well file.

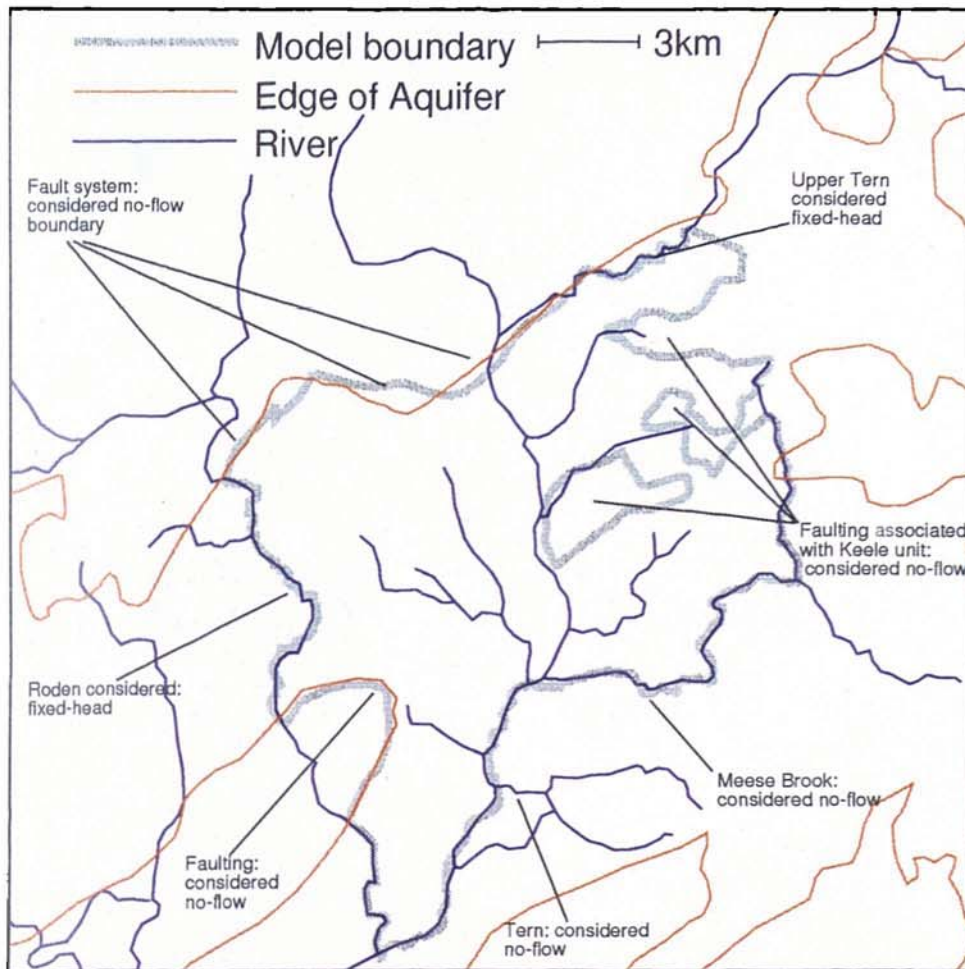


Figure 5.12: The location and nature of boundaries in the groundwater model.

groundwater system simulation this requires understanding the boundaries of the aquifer, the interactions with surface features, the spatial variability of aquifer properties and the interactions between all these features. Some of this knowledge will be translated directly into the numerical groundwater model, the rest adds to the understanding and insight into the behaviour of the system — valuable in assessing the performance of the model.

5.5.1 Delineating a groundwater unit

An initial task was to delineate some boundaries on the system. To a large extent the boundaries of SRA (1974) were taken as a template(see Section 5.2.3). Figure 5.12 shows the boundaries identified and used. The major differences are that the area covered is larger and, because the resolution is higher, the boundaries follow the natural features more closely.

5.5.2 Spatial distribution of hydraulic heads

The main groundwater body has a complex flow regime. One way of getting an indication of the major flows, and thus whether the boundaries chosen are reasonable, is by examining the spatial and temporal variation in watertable. It was also thought that such surfaces would provide good starting heads for the groundwater modelling exercises. The National Rivers Authority (NRA) provided several years of groundwater level readings from the monitoring borehole network. They also made available several contour maps of the watertable, interpolated by experts.

It was decided to use the groundwater readings, firstly because they were already in computer readable form, secondly because site data is more amenable to manipulation, and thirdly because it is possible to generate a surface for any period for which data is available. However, the contour maps are useful as they not only contain data but, perhaps more importantly, insights into the expert's opinion or own conceptual model of the groundwater flow regime.

The groundwater readings are taken at a network of monitoring sites, for the most part the sites have not been drilled specifically for this purpose — the network is thus patchy, Figure 5.13.

A standard method of generating a watertable surface

A standard method was developed to generate a watertable surface for any particular time, so that each surface would have similar properties and they could thus be compared. The first task was to extract a set of data from the readings database into the site file format of GRASS.

Very few of the groundwater levels were read on the same day, so there is a need for temporal as well as spatial interpolation. During non-pumping years most of the monitoring wells are sampled once a month, and so sets of readings were extracted for each month. Where a site had more than one reading in a month they were averaged. During periods when wells were being pump-tested or operated for river regulation, monitoring was much more frequent, so finding coincident readings was less of a problem. Groundwater levels outside but close to the groundwater unit, as delineated in Section 5.5.1, are also included in the interpolation.

Some of the readings cannot be considered valid for the purposes of interpolation, for example there are problems with readings near the Hawkstone Hill. Here the levels indicate an extreme mounding of the watertable. This was encountered by the SRA when they performed their modelling (SRA, 1974).

“ Early surveys of the groundwater levels in the pilot area appeared to indicate a recharge mound centred on Hawkstone Hill in the Northwest. Preliminary work for the first model simulation showed that such a recharge mound, in which groundwater levels rose 60 metres in a horizontal distance of 2km, was not possible unless either (a) the transmissivity of the Upper Mottled sandstone in the region was only 5 percent of that of the surrounding areas, or (b) the flow from the Hawkstone area was at least ten times the known recharge. Examination of the logs of 23 boreholes on Hawkstone Hill provided no evidence to support a zone of low permeability, but showed the aquifer to be hydrogeologically similar to the main aquifer west of the Hodnet Fault. ”

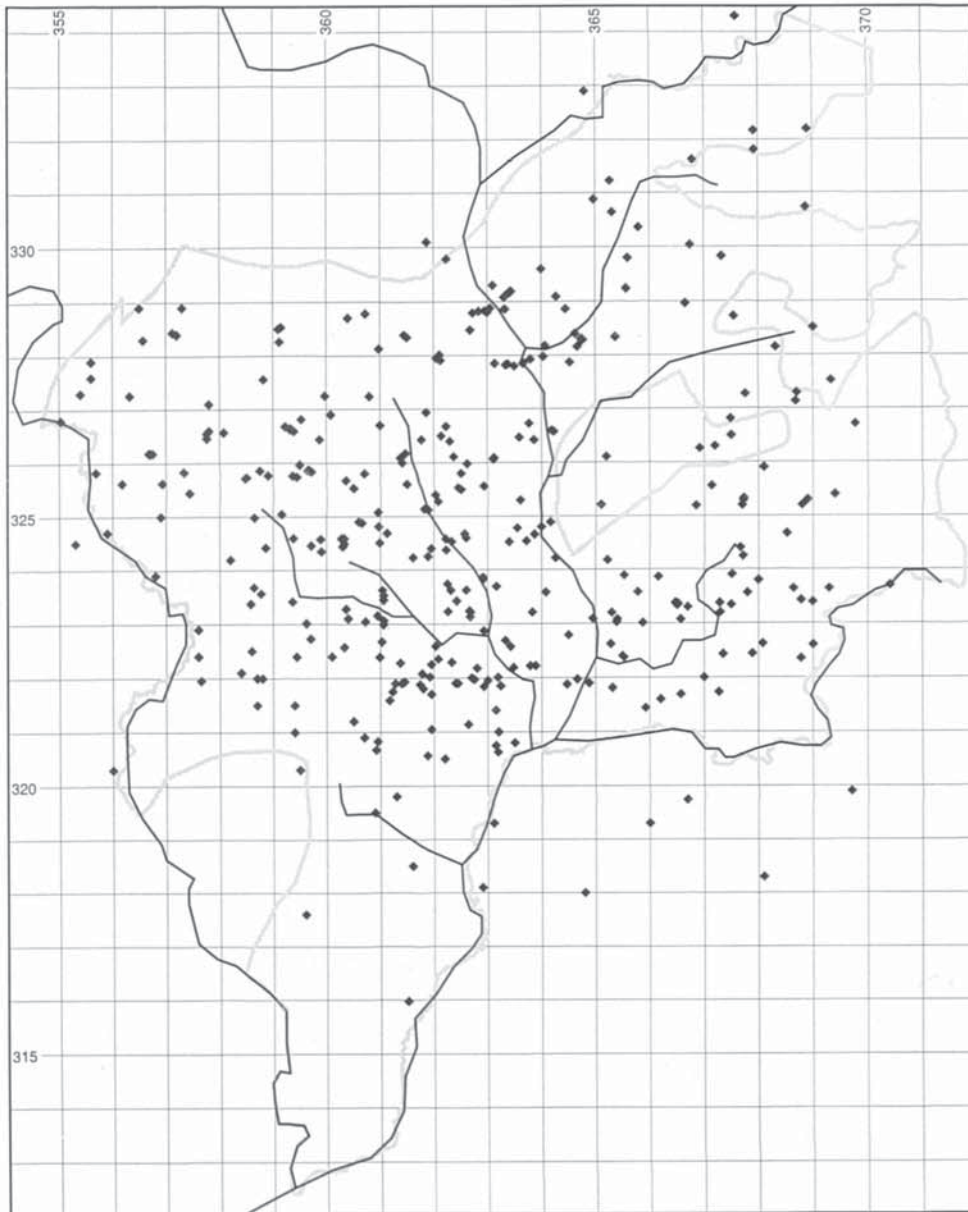


Figure 5.13: Tube wells and boreholes used for monitoring the watertable in the Tern region during the history of the SGS. The grid is given in kilometre squares.

The SRA scientists put the anomalous mound down to a perched watertable, for which there was some evidence. Those readings which were deemed to represent the perched aquifer were removed from the site file. Interestingly those sites identified by SRA (1974) as perched were later used by succeeding researchers to interpolate the main groundwater body watertable — a case of corporate memory loss!

For some of the modelling exercises it was necessary to produce a time-series of watertable surfaces. Here again problems arose. Some of the sites are not sampled each month, this means that some mounds or pits can appear in one month's interpolation which are not there in the subsequently interpolated months. One way to deal with this might be to linearly interpolate in time between the readings to get levels for intermediate months. However, often there was only one reading in a few months at the site, so when this occurred the readings were simply removed from the site files. This is thought acceptable as these sites were located well away from the area of interest.

There are two interpolators within GRASS that could have been used to interpolate the site data: *r.surf.idw*, which is an implementation of a distance weighted average algorithm and *s.surf.tps* which uses a spline approach (see Mitsova and Hofierka, 1993). The choice between the two is largely subjective. The spline method tends to give less jagged surfaces and has several parameters that can be changed to adjust the form of the resulting surface; *s.surf.idw* is more robust and far faster. Figure 5.14 shows sections across the Tern valley of interpolated watertable surfaces generated by both interpolators. In the end *s.surf.idw* was chosen because the manipulation described below caused *s.surf.tps* to behave very inconsistently.

An alternative option which was considered involved exporting the data to some external package and to use a geostatistical approach. This was again rejected, firstly because of the time that would be required, and secondly because the manipulations (described below) which were needed to make the generated surfaces consistent with the conceptual model of the groundwater system, would nullify the advantages of using a more statistically rigorous approach. However, a statistical and geostatistical analysis has been conducted to see if this might be an appropriate approach for future use — see Section 5.6.

Figure 5.15 shows an interpolation of the watertable based purely on the raw data extracted from the database. There are several ways in which this does not fit in with the conceptual model of the groundwater regime. There are a lot of pits and mounds. Mounds may be acceptable as they can occur in the groundwater unit, but pits are less acceptable and when the model is run the flow will tend to move into these pits and fill them. The map algebra facility *r.mapcalc* can highlight another problem. The interpolated surface is higher than the land surface in many areas, however, although it is known that there are some artesian areas in the Tern basin, if the DEM is subtracted from the surface as in Figure 5.16, the apparent artesian areas are much more widespread than known existing areas. The *r.mapcalc* statement to do this is:

```
r.mapcalc "artesian=if(dem<=watertable,watertable-dem,null())"
```

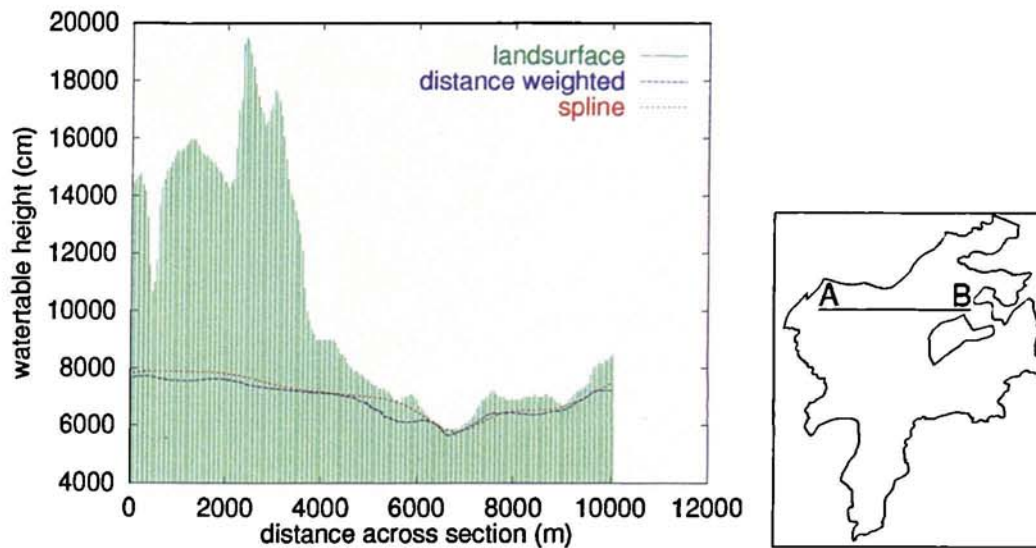


Figure 5.14: Profiles of the March 1990 watertable generated with different interpolators. The profiles extend from A to B on the location map.

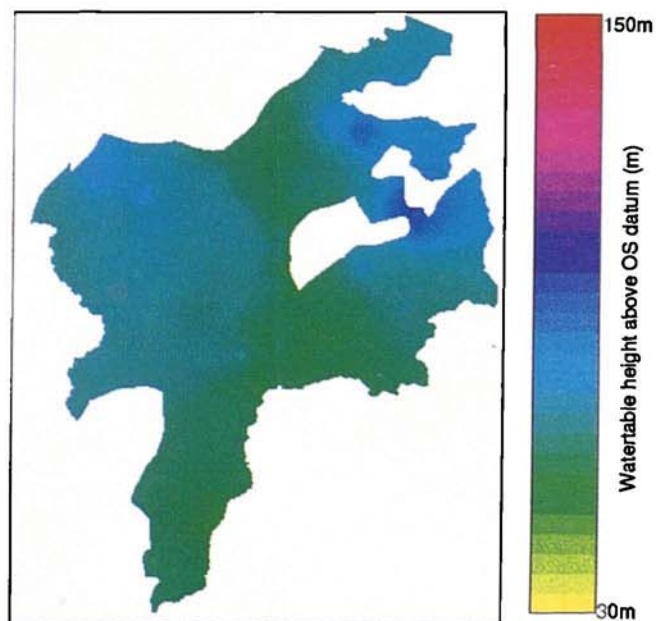


Figure 5.15: Interpolated watertable for March 1990, using *s.surf.idw*.

Inspection of Figure 5.16 shows that the artesian areas often coincide with the river network. As explained in Section 5.5.1 some of the rivers act as “fixed-head” boundaries and thus the hydraulic head in the river is a reflection of the hydraulic head in the aquifer at these boundaries.

To force the interpolator to respect those boundaries interpreted as fixed-head, it was necessary to add artificial groundwater readings to the GRASS site file to include this knowledge of the connection between the river and the groundwater. For the purposes of this model a value of $3m$ below the land-surface as indicated by the OS DEM was adopted as the river level. The OS DEM has too coarse a resolution ($50m$) to detect most river channels in the area and from observations of the rivers of the region it was noted that they tend to incise about $2-3m$ below the land surface; see for example Figure 5.17. These are admittedly imprecise figures but considered of adequate accuracy for the purpose here.

The method adopted was to subtract $3m$ from the DEM. Areas not designated as fixed-head boundaries were masked off and then the specially written program *r.to.sites* was used to convert the remaining areas to GRASS sites. Finally the UNIX *cat* was used to join the river sites file to the groundwater readings site file and the resultant site file is then interpolated using *s.surf.idw*. This is achieved with the following sequence of GRASS, UNIX and specially written commands:

```
r.mapcalc "fixed_boundaries=if(model_bound2==-12,dem-3,null())"
r.to.sites map=fixed_boundaries sites=fixed_boundaries -nh
cat water_table fixed_boundaries > water_table_fb
s.surf.idw input=water_table_fb output=water_table_fb
```

On examination of the surface in Figure 5.18 and the artesian areas in Figure 5.19 wide spread artesian areas can still be seen, largely around the Tern. This is because although the Tern is not being treated as a fixed-head boundary (Section 5.5.1) there is still considerable interaction between it and the aquifer, thus this knowledge was included in the interpolation in the same way as described above. Figure 5.20 shows the final surface, and Figure 5.21 the calculated artesian areas.

Figure 5.22 shows profiles of the watertable when adding these different assumptions.

5.5.3 Groundwater flows

All the interpolated surfaces and the professional contour maps supplied by the NRA (now the Environment Agency) broadly indicate the major gradients and thus the flows of the groundwater system. Figure 5.23 shows the major flow directions as interpreted from these maps. Knowledge of the major groundwater flows gives clues to many characteristics of the groundwater regime, the nature of the boundaries and the recharge and loss. This knowledge also help in the statistical and geostatistical investigation of the system — see Section 5.6.

² -1 is the code used by Modflow to indicate a fixed-head boundary.

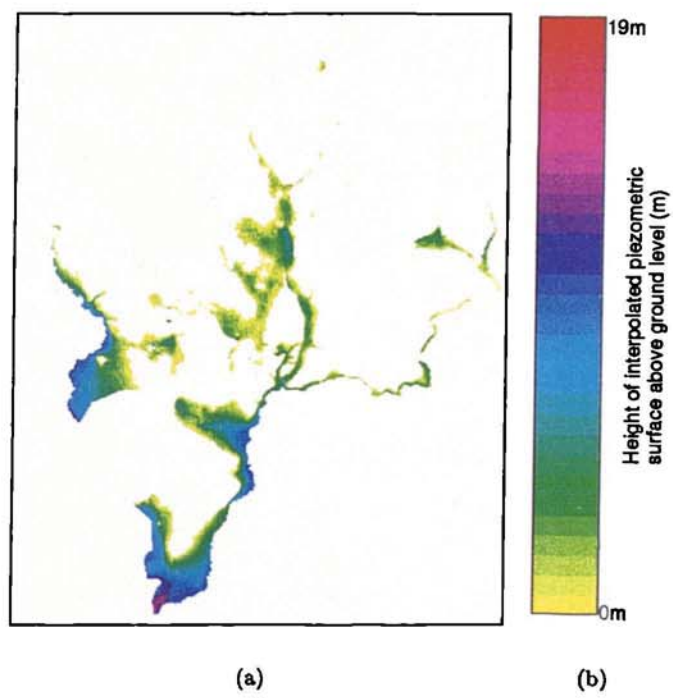


Figure 5.16: Areas that appear artesian, when March 1990 data is interpolated with *s.surf.idw*, and subtracted from the DEM.



Figure 5.17: The River Tern near Stoke on Tern.

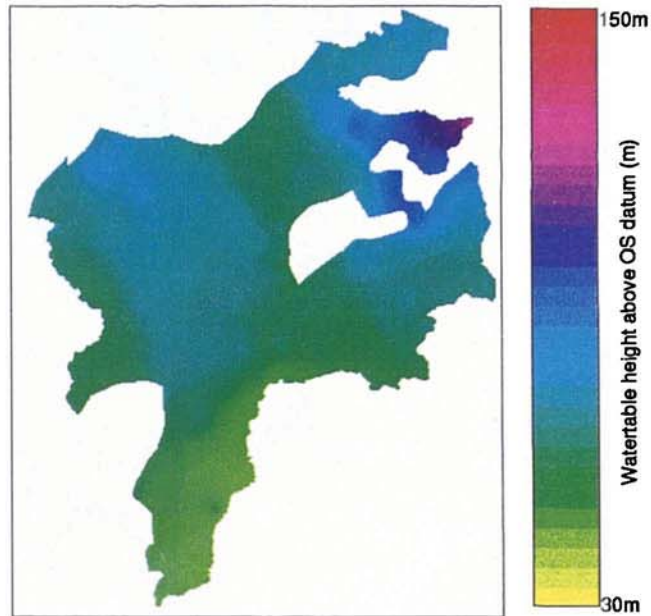


Figure 5.18: Interpolated watertable for March 1990 including fixed-head boundaries.

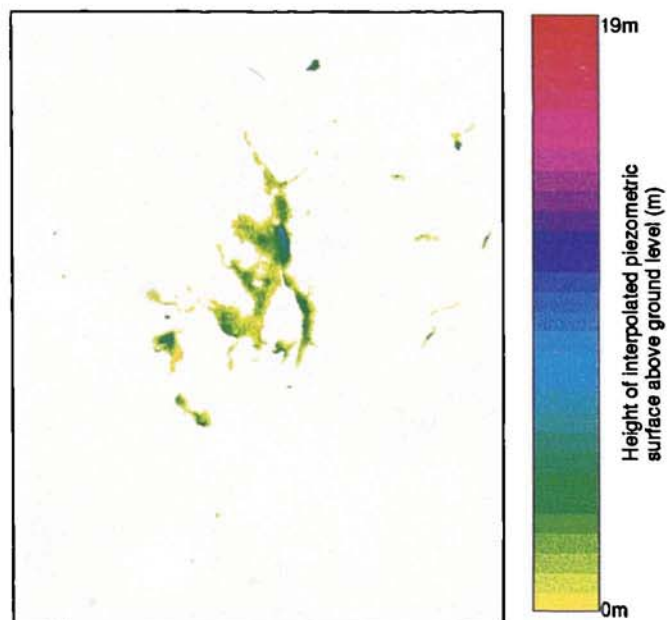


Figure 5.19: Interpolated artesian areas, accounting for fixed-head boundaries within the interpolation.

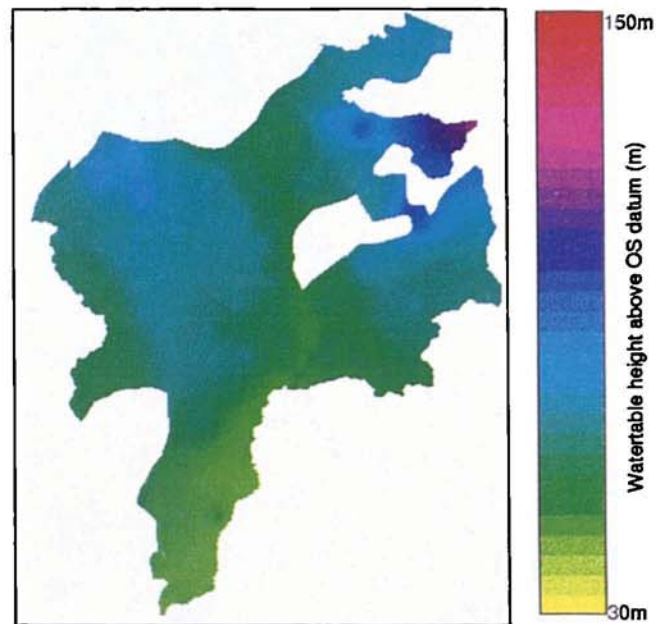


Figure 5.20: Interpolated watertable for March 1990, accounting for fixed-head boundaries and the Tern river.

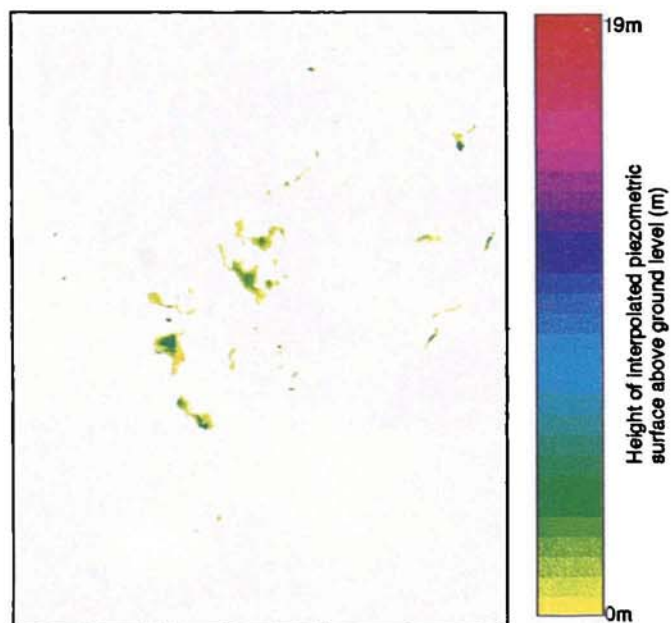


Figure 5.21: Interpolated artesian areas, accounting for fixed-head boundaries and the Tern river.

5.5.4 Transmissivity

Characterising the transmissivity of the aquifer proved problematic, as the SRA (1974) discovered:

“ Seven reliable pumping test results are available for the aquifer... . This is a rather limited amount of data and one of the reasons for adopting a single layer model; at this stage, and with data available, a multi-layered model would not be justified. ”

One method examined by the author was to define transmissivity regions, that is, regions across which the transmissivity might reasonably be taken as constant (Rushton and Bishop, 1993, termed these *transmissivity zones*, see Section 5.2.2). Aquifer depth and fault zones were used to divide the region up, however no correlation could be found between physical interpretation of the aquifer and the transmissivities observed (Figure 5.24). This can be due to a number of complicating factors, inappropriate pump test analysis, partial penetration of the well (Figure 5.25), complex geology. In the end little could be done to improve on the selection of a constant value across the aquifer. The values in Figure 5.24 give an indication of the sort of values that might be appropriate but some sort of calibration process is inevitably required — see Section 5.8.

5.5.5 Storage

Storage proved as problematic as transmissivity. As in the model described by SRA (1974) two values were used one for confined and one for unconfined conditions, no further resolution was possible, the values in Figure 5.26 give an indication, but again some calibration is inevitable — Section 5.8.

5.5.6 Recharge

Rushton and Bishop (1993) when modelling the Nottingham Aquifer, made the assumption that there was no recharge when there is a soil moisture deficit (see Section 6.1.2 for an explanation of soil moisture deficit). They also modified recharge on the basis of land use. A similar set of assumptions can be made in the Tern area. This has the effect that recharge is virtual zero in most summers. During the winter it appears that it is the distribution of clay deposits that controls the spatial variation of recharge. Simple cartographic modelling can be used to generate rasters of recharge, knowing the distribution of clay drift and the rainfall for the period that the model is to simulate.

5.5.7 River leakage

Defining river leakage is problematic, during the course of the pilot investigation experiments and desk studies were performed by the Severn River Authority and later by the Severn Trent Water Authority to try and quantify this process. This was done in order to get an estimate of the efficiency of the Scheme, i.e. the proportion of the water pumped from the Tern that actually

contributes to an increase in river flow rather than being immediately returned to the aquifer under the influence of the increase in gradient between the river head and the aquifer head caused by the pumping itself.

The most intensive of these experiments occurred in 1976, it was an investigation into the relationship of river stage to the groundwater level. This was performed at the Helshaw Grange pilot borehole, and involved pumping the borehole (STWA, 1978), the investigators concluded:

“ ... that this short period of pumping did not reverse groundwater gradients and therefore storage was not developed beneath the river. The river is now thought to act as a partial recharge boundary but actual leakage would only occur after a far greater pumping period than carried out for this short test, estimated to be at least 100 days.”

The investigation was extended by testing the permeability of river bed sediments at 26 sites using a falling head test technique. The survey found a large variability in permeability which was attributed to the “highly variable nature of river sediments”. Low permeability seemed to be related to peat and silt.

“ It was concluded that an average leakage value of $1.5 \text{ Ml.d}^{-1}.\text{km}^{-1}$ was appropriate for the River Tern once the effects of lateral leakage through the river banks had been considered. ”

On test pumping the Helshaw Grange borehole in 1984 (NRA, 1985) more evidence was found concerning the Tern's interaction with the aquifer:

“ The difference in drawdown between the Heathbrook East and West boreholes is significant as they are only 20m apart but are separated by the River Tern. These differing effects confirm the suggestion that the river acts as a form of partial recharge boundary. ”

5.5.8 Confinement

As noted briefly in Section 5.1 aquifers behave differently if they are confined. Though over large areas the aquifer outcrops at the surface, there are significant deposits of boulder clay elsewhere that can act as a confining layer. Figure 5.27 shows the drift geology of the area as envisaged by Skinner (1979) in his presentation to the Public Inquiry. During the pump testing of the Hodnet Station in 1984 (NRA, 1985) it was noted that :

“ ...the confined nature of the aquifer to the south ... results in an asymmetric cone of influence with rapid effects being noted over 1km from the site. ”

And during the Helshaw Grange pump test it was reported:

“ Drawdowns were also observed in the Wollerton borehole. This site is 1.5km from the abstraction borehole and had not been drilled when the previous test took place. The effect indicates that the cone of influence is substantial and may spread preferentially along the river valley where confined conditions exist. ”

Modflow can handle the change between confined and unconfined conditions the occurs as the piezometric surface varies temporally and so this does not present too significant a challenge.

Given that the general difference in magnitude between confined and unconfined storage coefficients is usually very large compared with spatial difference of either confined or unconfined storage. It is likely that ensuring the change from confined to unconfined and back occurs at the correct time during a model run, by ensuring the relative levels between the watertable and any confining strata are correctly modelled, will have more influence on the success of the modelling than ensuring the spatial distribution of either confined or unconfined storage are accurately known. Or to put it simply it is of little use having the unconfined storage at a point accurately known if at that point the model, if it were simulating the real world adequately, should be modelling confined conditions. See Section 5.8.2 for one way in which this change in conditions might be handled using a cartographic modelling approach.

5.5.9 Fault zones

Faults have been discussed in Section 5.5.4, as divisions between units, but a fault is also a zone in its own right and has its own varying hydrogeological properties. They can be regions of very high or very low conductivity often associated with fracture flow and can cause anisotropic conditions. In this area there are known to be several large faults crossing the region. Some have been taken as "no-flow" boundaries as they cause the aquifer to abut relatively impermeable rock. However, how do those that cross the groundwater region behave?

A second issue to resolve is that many geological features are inferred from indirect evidence; e.g. some other faults may exist but are unknown. Figure 5.28.a shows the faults as envisaged in 1963, the time of the last revision to the geological map of the region³. During the course of the Shropshire groundwater investigation and the scheme proper more light has been thrown on geology of the area, Figure 5.28.b shows the new arrangement of faults envisaged after the 1984 pump tests (NRA, 1985).

" The unexpected effects resulting from the Hopton test suggest that the geology of the area is considerably more complex than indicated in the published map [BGS map sheet 138]. In particular, to explain the apparent independence of the Hopton and Lodgebank areas, a barrier boundary, presumably a fault, must occur between the two sites. "

Test pumping in 1984 provided some information about the way the faults in the region act hydrologically (Section 5.8.2). Further away from the test pumping the behaviour of faults can only be inferred or estimated by recalibration the model. Figure 5.29 shows the hydrogeologist's interpretation of observed drawdown (NRA, 1985).

As an initial step to include faults within a groundwater model, they can be treated as no-flow or partial-flow barriers. The geological map can be taken as an indication of where the faults

³ British Geological Survey, sheet 138.

may be, but it should not be followed slavishly. Calibration will be necessary, but how this is to be performed systematically when it involves not only calibrating the hydrological properties of the faults, but also their position and the properties and positions of postulated faults, is unclear

in Section 5.8.1 faults are included as partial no-flow boundaries and the effects of different transmissivities is investigated.

5.6 Statistical investigation of the watertable

A significant component in the interaction between the groundwater and the soil moisture is the position of the watertable. One of the main reasons for adopting a dynamic modelling approach as opposed to the static approach adopted by Hedges (1989) was because the watertable is expected to have natural fluctuations, as stated in Section 2.3. The watertable is a surface that varies in time but it is one that is hidden and can only be observed at discrete spatio-temporal points. The purpose of this section is to use statistical and geostatistical techniques on the available data to better understand the properties of the surface and its temporal variation. Questions that must be resolved before numerical modelling can begin are:

- is there an observable recession in the groundwater levels during a year and what is its nature?
- does the river have a role in the groundwater regime?
- can the distribution of groundwater levels be predicted statistically?
- if so what independent variables are important?

Watertable monitoring data was provided by the National Rivers Authority now the Environment Agency . The dataset is very large and covers several years of groundwater observation at hundreds of sites, often taken as frequently as once a month. Figure 5.13 shows all the monitoring boreholes and tubewells used throughout the history of the Groundwater investigation and the scheme. Many of the boreholes were sunk for purposes other than monitoring and when they were sunk for monitoring it was often for the localised effects of nearby pumping rather than monitoring the whole groundwater regime. However, the readings are a valuable source for building a picture of the groundwater regime

5.6.1 Recession

It was contended in Section 2.3 that natural fluctuations in the watertable would be a complicating factor in any analysis. In particular a marked recession in particularly dry years (just those years when pumping would occur) was expected. On examination of the hydrographs of many of the wells, as in Figure 5.30, there does appear to be a general yearly recession a recession that is

Month	Dataset	Mean	Standard Deviation	Minimum	Maximum	Number
March 1990	all watertable heights	67.962	8.118	53.750	105.600	140
	heights without outliers	66.755	6.392	53.750	96.590	130
	all watertable depths	6.267	9.513	-2.970	67.840	140
	depths without outliers	5.760	8.900	-2.970	67.840	130
June 1990	all watertable heights	67.117	7.262	53.320	96.380	129
	heights without outliers	66.434	6.447	53.320	96.380	123
	all watertable depths	6.267	8.883	-2.630	67.780	129
	depths without outliers	5.647	8.864	-2.630	67.780	123
Sept 1990	all watertable heights	67.0109	7.2360	52.9300	95.3300	127
	heights no outliers	66.4341	6.4917	52.9300	95.3300	122
	all watertable depths	6.6457	8.9366	-2.4700	67.9200	127
	depths without outliers	6.4903	8.9086	-2.4700	67.9200	122

Table 5.1: Univariate statistics for watertable data.

more obvious in some wells than others. Some simple statistical tests were performed to see if an overall yearly recession was detectable. Well records for each month in 1990 were extracted from the *RDB* database⁴. Where a month had more than one record these were averaged for that month. Table 5.1 gives the univariate statistics for all the sets and subsets of data examined here. The means appear to indicate a very slight drop in watertable during the summer months.

Can the mean be taken as an appropriate measure of location of the watertable? Figure 5.31.a shows the basic histogram of observations for March 1990 and Figure 5.31.b shows the histogram of depths for March 1990. The watertable appears to exhibit some normal-distribution characteristics and a normal-probability plot (Figure 5.31.c) confirms this. However, depth is definitely not normally distributed (Figure 5.31.d). Data for September shows similar properties (Figure 5.32.a, b, c and d) to those of March. This can be explained by the way the network of observation wells has been built, as stated above many of the boreholes were sunk for purposes other than monitoring the total groundwater regime. Thus there will be more wells in areas where groundwater is easily obtainable, i.e. areas where the watertable is close to the surface, and more wells in areas where water is required, i.e. arable land, communities and river regulation sites. Thus we can expect a bias towards flat low lying areas. The more important factor appears to be closeness to the surface, indicated by the greater departure from normality in the watertable depth than the watertable height data.

A simple paired one tailed t-test was performed to compare the raw watertable height data

⁴ *RDB* is a freely available set of relational database operators, (Hobbs, 1993), its use is described in Appendix A.5. The principles of relational databases are described in Avison (1992).

Months of readings	Degrees of freedom	t-statistic	Significance
March and June	125	5.67	0.1%
March and September	123	7.25	0.1%
June and September	117	6.08	0.1%

Table 5.2: Results of t-test on raw watertable height data from 1990. All tests are significant at the 99.9% level.

between March and June, March and September and between June and September. The results are in presented Table 5.2, and indicate that there was been a significant movement in the mean an annual recession.

However, despite the above analysis there are still some doubts about the normality of the watertable height data, although it appears normal in the plots, the way the monitoring network came about casts doubts upon this, and the watertable depth data is certainly not normal. Therefore a Wilcoxon test was performed, and this also revealed a significance movement in the central location of the distributions, Table 5.3.

Some of the data are unreliable and were removed, these included several boreholes which are located in the Hawkstone region. A region which has been identified as a possible area of a perched watertable, see Section 5.5.2. Other data are removed because they are contradictory to nearby data points, indicating unlikely groundwater gradients or more probably that they are isolated from the main groundwater body. When these are removed and the tests performed again the results are the same as above (see Tables 5.4 and 5.5).

It is concluded from the above analysis that the mean is a good statistic for identifying movement in the watertable. Given this assumption then Figure 5.33 illustrates and summarises the recession in 1990.

5.6.2 Statistical investigation into the effect of river location on the watertable surface

In Sections 5.5.2 and 5.5.7 it was postulated that the rivers of the region play a significant role in the hydrogeological regime. If this is the case then different statistical properties would be expected of the watertable surface at differing distances from the river. To investigate this the data, without outliers, were divided into subsets based on distance from the river. In order to get a distance value associated with each data point a map was created using the GRASS command *r.buffer*, Figure 5.34 ⁵.

As expected the mean watertable height increases with distance from river, though there is an

⁵ The *RDB* script *av_wt_per_month* described in Appendix A.5 was used to extract the height data and format it for the statistical package.

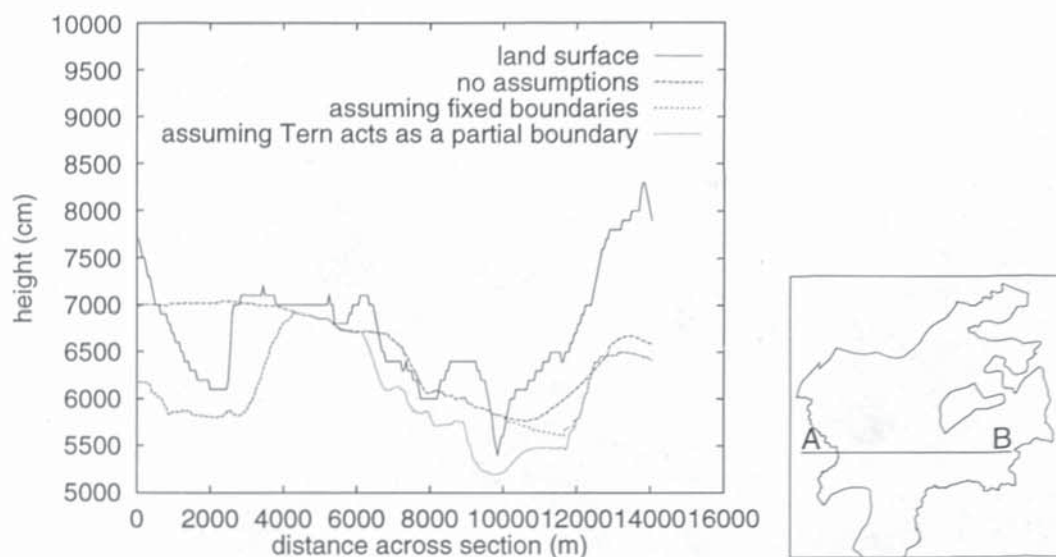


Figure 5.22: Profiles of the March 1990 watertable, generated with differing assumptions, profiles extend from A to B on the location map.

Months of readings	Nos. pairs	Z	Significance
Watertable heights			
March and June	126	-7.3165	99%
March and September	124	-7.4453	99%
June and September	118	-7.6127	99%
Watertable depths			
March and June	126	-7.3214	99%
March and September	124	-7.7570	99%
June and September	118	-7.9228	99%

Table 5.3: Results of Wilcoxon test on raw watertable height data from 1990. All tests are significant at the 99% level.

Months of readings	Degrees of freedom	t-statistic	Significance
March and June	119	8.21	99.9%
March and September	118	8.25	99.9%
June and September	113	5.79	99.9%

Table 5.4: Results of the t-test on watertable height data from 1990 with outliers removed. All tests are significant at the 99% level.

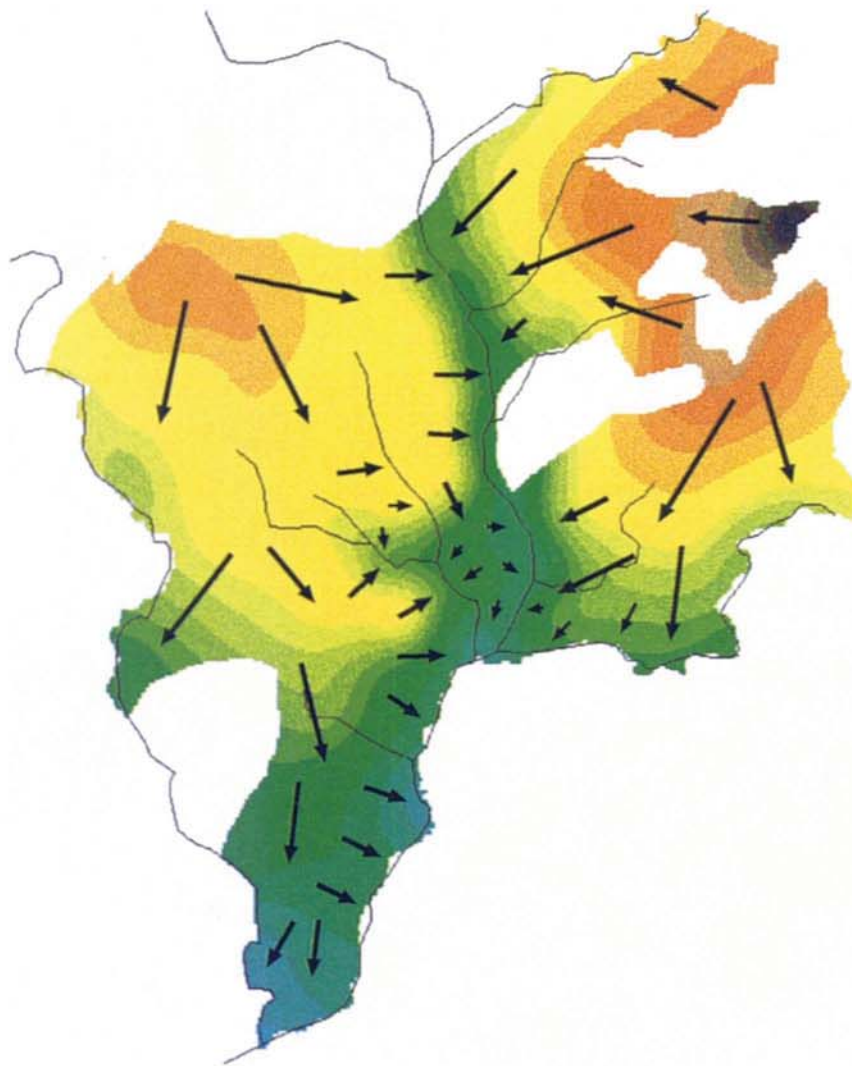
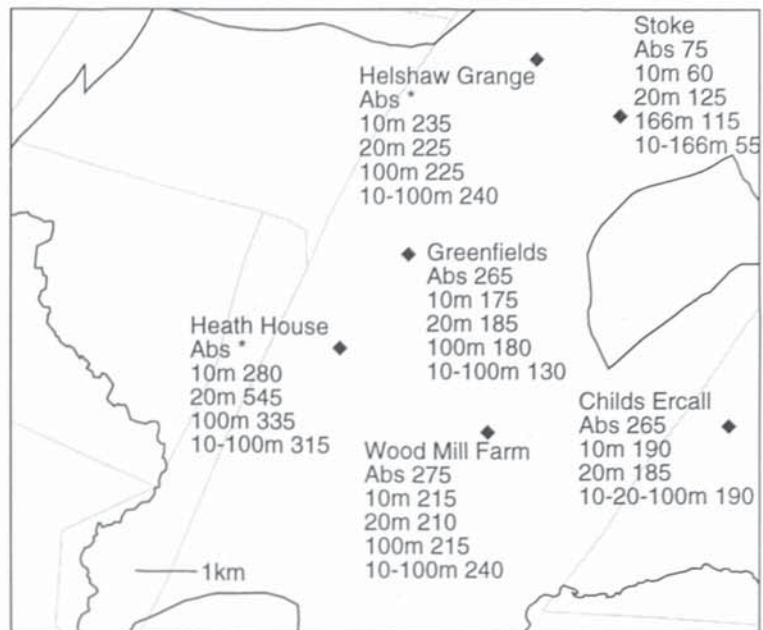
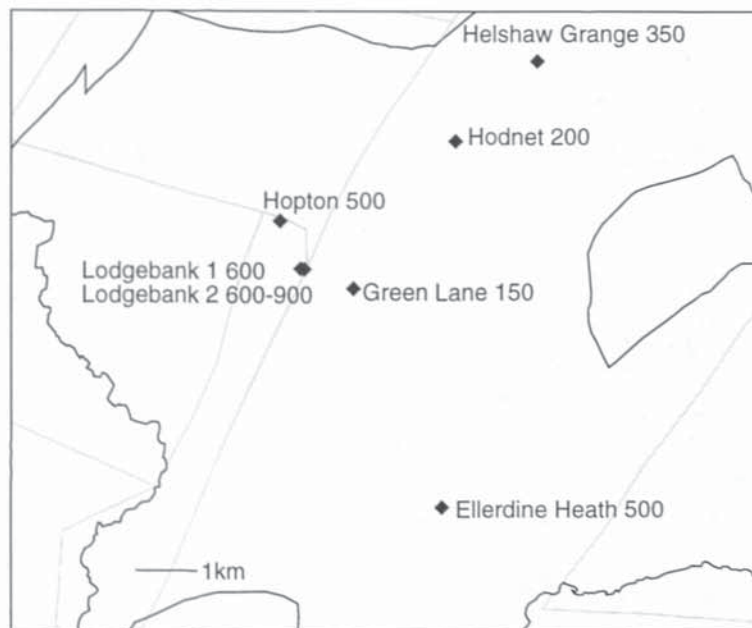


Figure 5.23: Major flow directions of the groundwater system.



(a)



(b)

Figure 5.24: Transmissivities from (a) 1973 pump tests (SRA, 1974) and (b) 1984 pump tests (NRA, 1985). Grey lines show how the aquifer was divided into regions at faults zones.

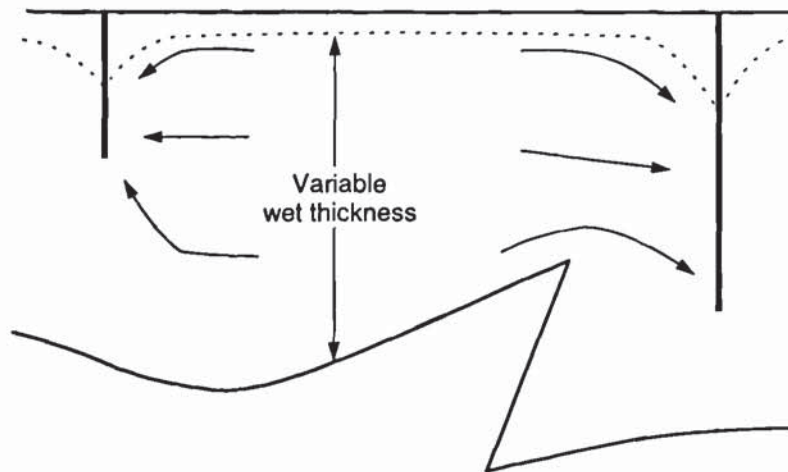
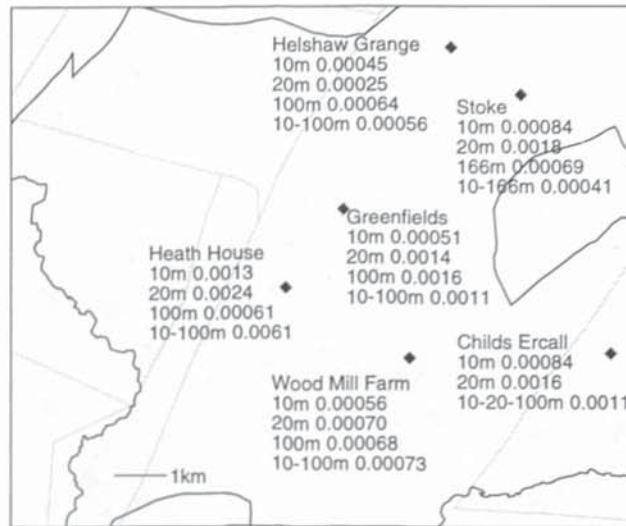


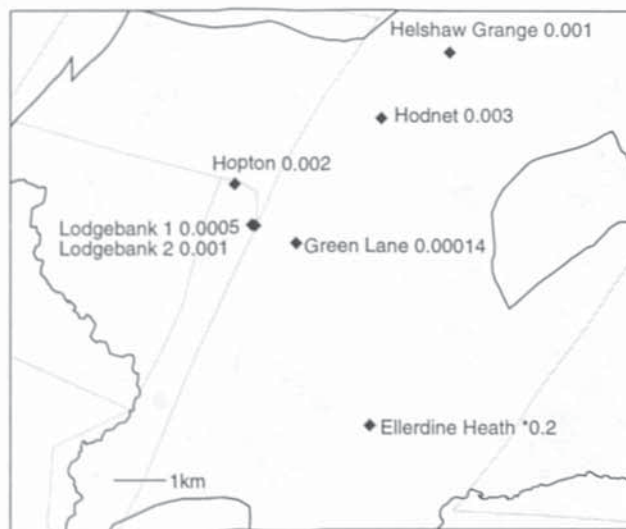
Figure 5.25. The effect of partial penetration of wells. The transmissivity derived from partially penetrating well may lead to effective transmissivity parameters that are different when one is considering pumping scenarios and general groundwater movement.

Months of readings	Number of pairs	Z	Significance
Watertable heights			
March and June	120	-7.3577	99%
March and September	119	-7.4971	99%
June and September	113	-7.3696	99%
Watertable depths			
March and June	120	-7.3630	99%
March and September	119	-7.8181	99%
June and September	113	-7.6860	99%

Table 5.5: Results of the Wilcoxon test on watertable height data from 1990, with outliers removed. All tests are significant at the 99% level.



(a)



(b)

Figure 5.26: Confined storage coefficients from (a) 1973 pump tests (SRA, 1974) and (b) 1984 pump tests (NRA, 1985). * considered particularly unreliable by the analysts. Grey lines show how the aquifer was divided into regions at faults zones.



Figure 5.27: Drift in the Tern area, after Skinner (1979).

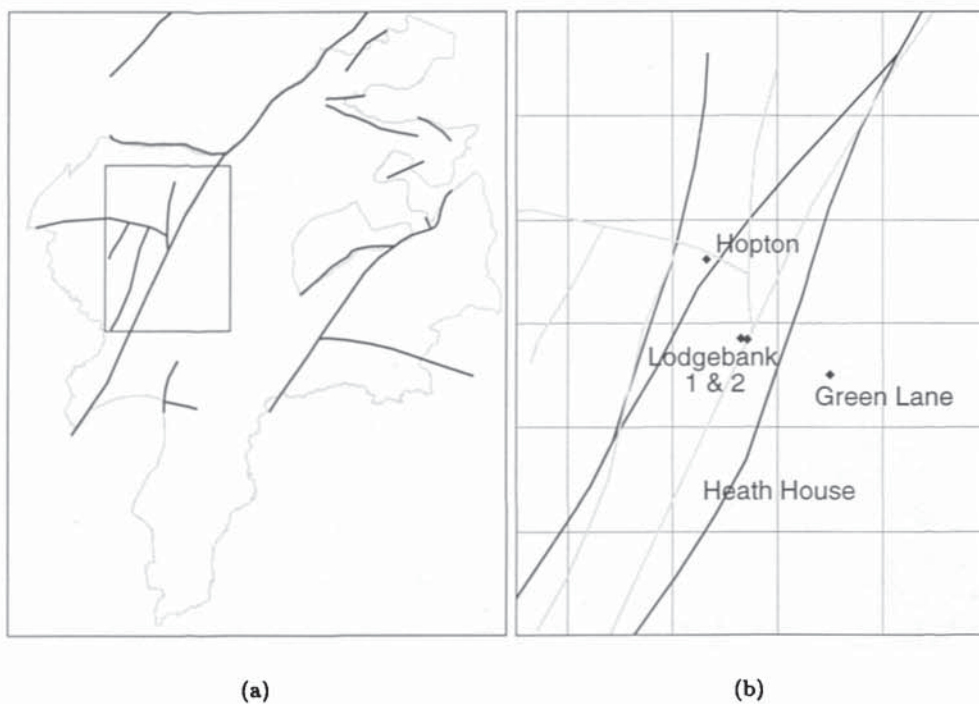


Figure 5.28: Faults of the Tern area (a) as envisaged in 1963, (b) revised interpretation after the 1984 pump tests (NRA, 1985). The Geological Survey's published maps have yet to be revised.

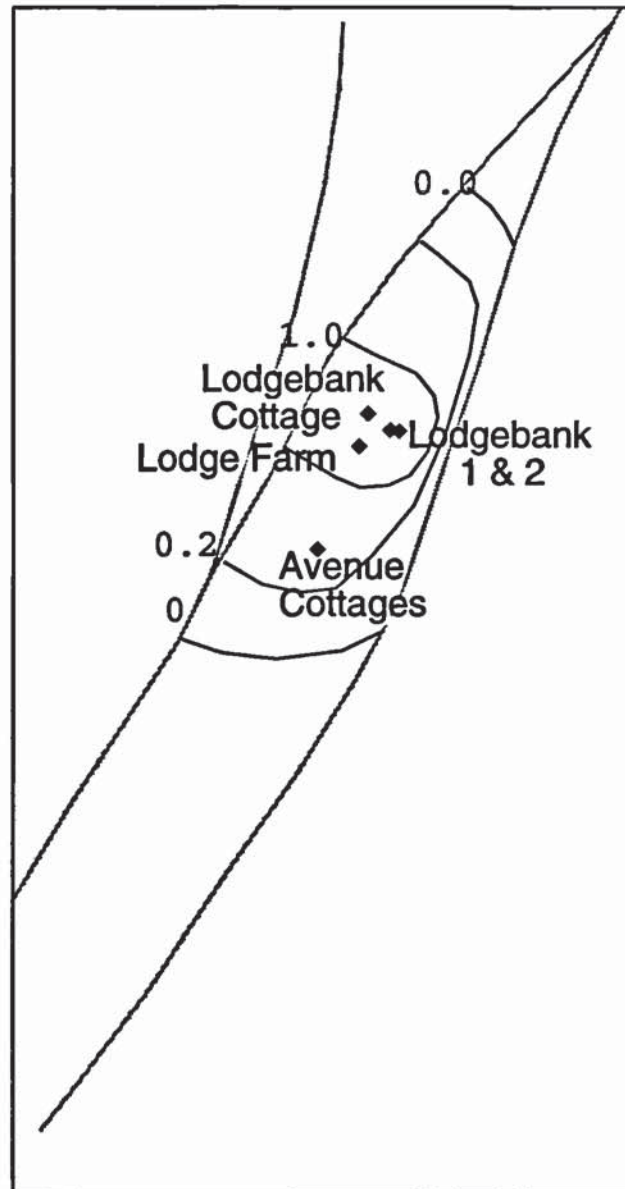


Figure 5.29: The effects of faulting on the Lodgebank pump test, 1984. The watertable after 22 days NRA (1985), the eastern fault is interpreted as acting as a partial barrier, the western fault acts as a near total barrier.

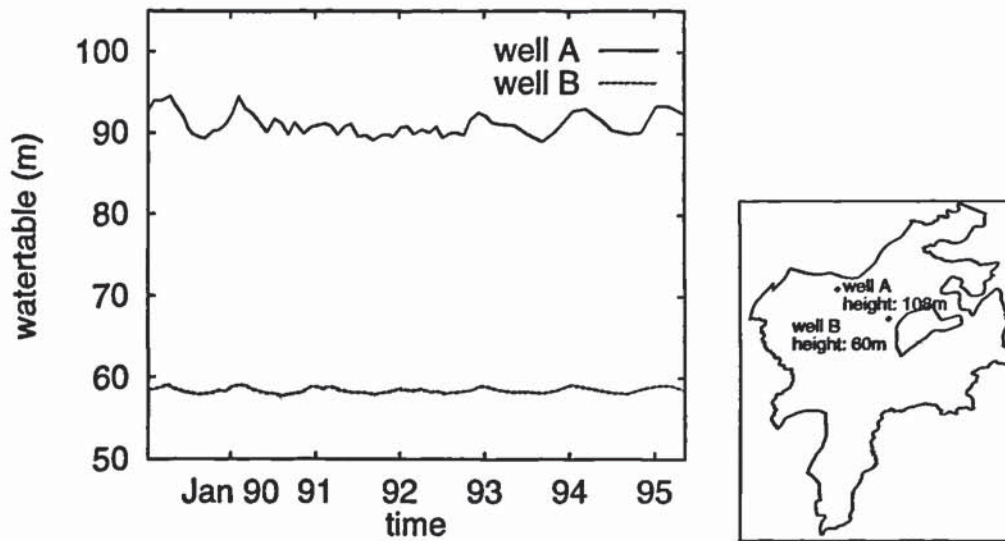


Figure 5.30: Typical hydrographs at two monitoring wells, 1989-95.

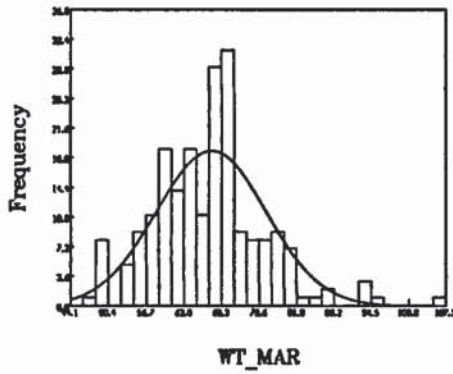
Distance of readings	Number of samples in January	Mean in January	Standard deviation in January
<500m	47	61.51	4.71
500-1000m	24	67.00	2.99
1000-1500m	12	68.35	4.59
1500-2000m	12	72.00	5.98
2000-2500m	13	70.59	1.56
>2500m	9	75.47	8.09

Table 5.6: Univariate statistics of watertable heights in 1990 grouped by distance from river.

unexplained reversal of this trend between those readings at 1500–2000m from the river and those between 2000–2500m. Unexpectedly the apparent annual recession of the watertable does not increase with distance. The situation initially envisaged was that watertable would relax during the course of a year something like Figure 5.36.a but the case appears to be, with this preliminary investigation, more like that of Figure 5.36.b.

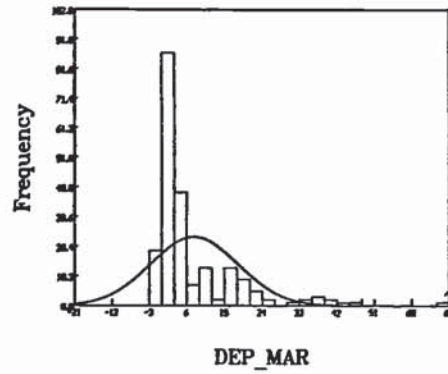
This conclusion should be treated with caution. The groundwater system is very complex and, as stated in Section 5.5.7, the interaction between the rivers of the region and the aquifer is not uniform along the length. Remember also that samples included within the same class on the basis of distance from river can be from very different areas of the region. Furthermore the number of samples in the groups becomes very low as they are split up on the basis of distance (e.g. Table 5.6)

Frequency distribution of WT_MAR



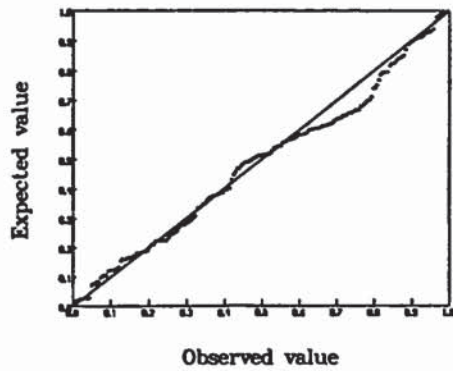
(a)

Frequency distribution of DEP_MAR



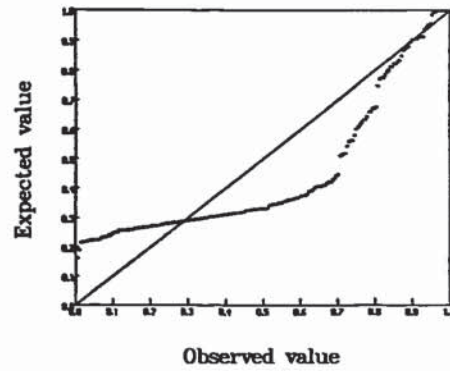
(b)

Normal probability plot of WT_MAR



(c)

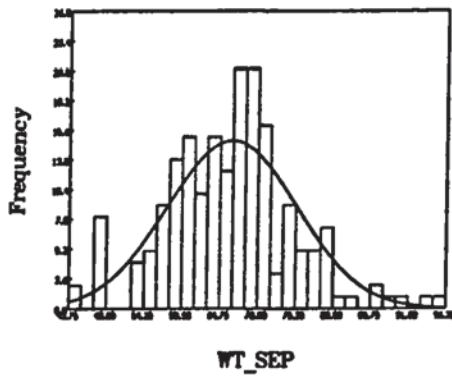
Normal probability plot of DEP_MAR



(d)

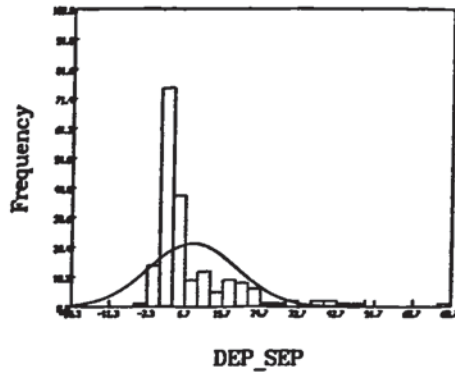
Figure 5.31: Histograms of (a) watertable height data and (b) watertable depth data, and normality plots of (c) watertable height data and (d) watertable depth data, for March 1990.

Frequency distribution of WT_SEP



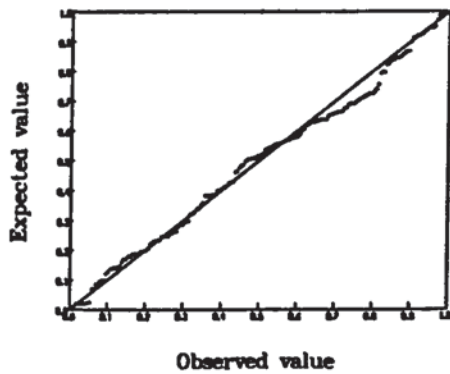
(a)

Frequency distribution of DEP_SEP



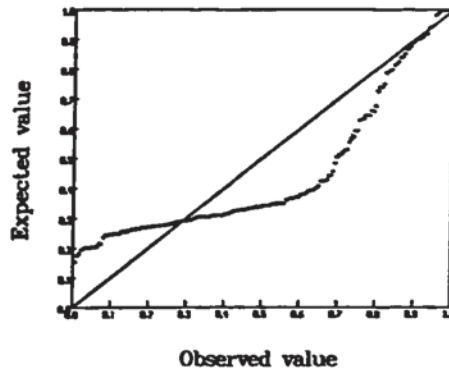
(b)

Normal probability plot of WT SEP



(c)

Normal probability plot of DEP_SEP



(d)

Figure 5.32: Histograms of (a) watertable height data and (b) watertable depth data, and normality plots of (c) watertable height data and (d) watertable depth data, for September 1990.

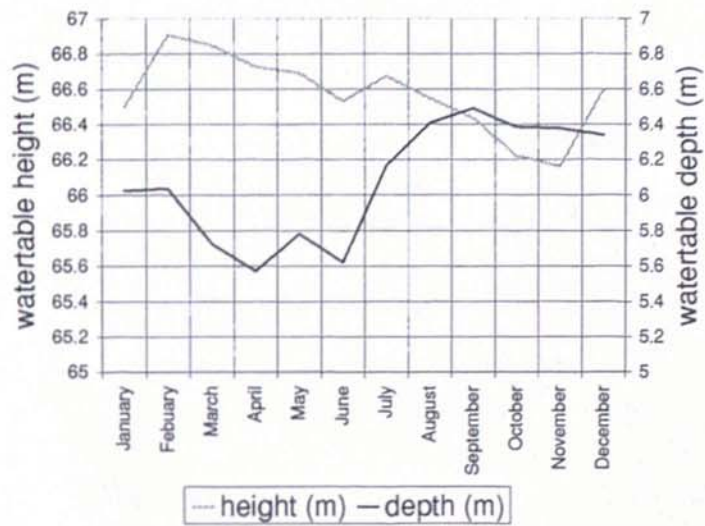


Figure 5.33: Variation of mean watertable heights and depths through 1990.

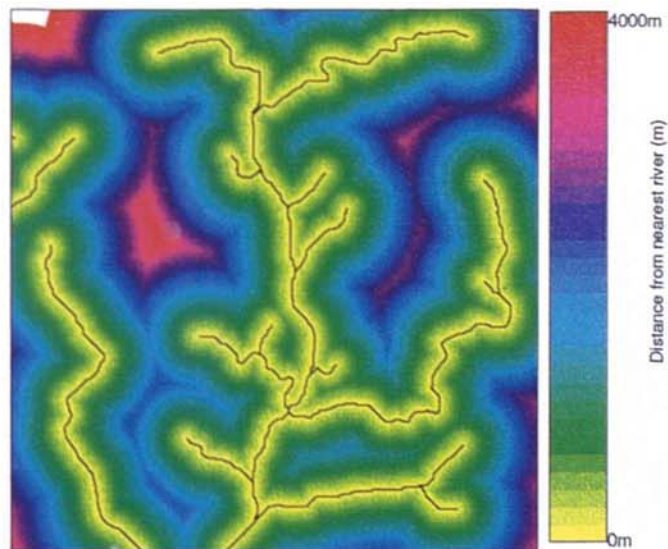


Figure 5.34: Buffer zones around the river network.

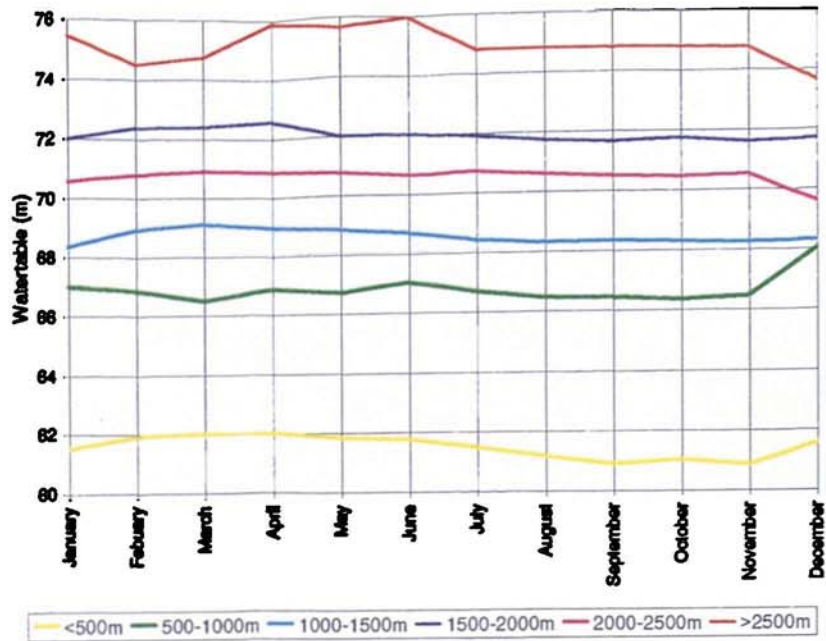


Figure 5.35: Variation in mean watertable height with distance from river in 1990.

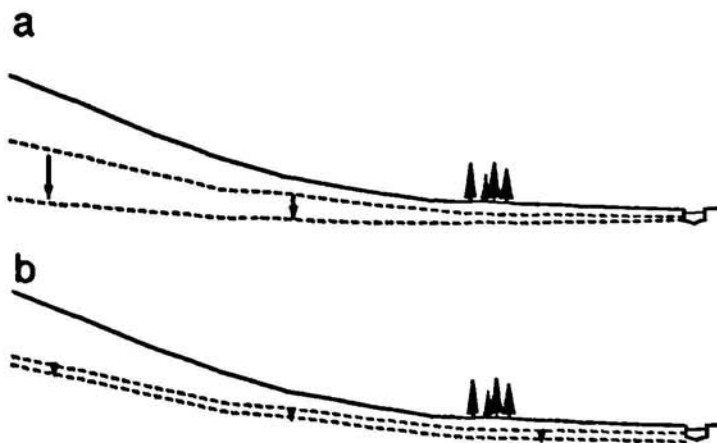


Figure 5.36: Different hypotheses of the annual variation of the watertable. (a) The recession is more pronounced in the higher watertable areas and (b) the recession is more even of the region.

5.6.3 Statistical prediction of the watertable surface

In Section 5.5.2 interpolated fields of groundwater heads were generated. At that time several drawbacks in the approach adopted were noted, and the desirability of looking for alternative approaches was stated. As the data had been conveniently extracted and formatted for the statistical package, it was decided to explore the possibility of using statistical relationships to make predictions of the position of the watertable.

Clarke (1994) suggested a general form for a statistical model of watertable distribution (Equation 5.9).

$$Wt_s = f(Elev_s, Slope_s, Dist_s, \dots; \theta) + \epsilon_s \quad (5.9)$$

where:

- s — is the spatial domain of the function;
- Wt_s — watertable depth variable in the spatial domain s ;
- $Elev_s$ — land surface elevation;
- $Slope_s$ — slope variable;
- $Dist_s$ — distance from river variable;
- θ — set of fixed parameters estimated from data;
- ϵ_s — random component representing fluctuations about systematic component.

Clarke leaves the function open ended indicating that this is by no means an exhaustive set of explanatory variables. A similar function could be written for watertable height, and for this investigation five explanatory variables were assembled.

- Land surface elevation (DEM) — extracted from the Ordnance Survey Landform Panorama dataset. This is a raster dataset giving the land surface elevation to the nearest metre with 50m resolution and a height accuracy of 3m.
- Distance from river — the river network was derived from the DEM using the GRASS command *r.watershed* (Westervelt et al., 1993) and converted to distance from river using *r.buffer* command with 50m resolution.
- Landscape position index (LPOS) — LPOS is an index suggested by Fels and Matson (1996). It is designed to place a pixel in the context of the surrounding landscape structure — the more positive the index is, the closer to the bottom of a valley the pixel is. The equation for landscape position is:

$$LPOS = \frac{\sum i = n \frac{(E_i - E_0)}{d_i}}{n} \quad (5.10)$$

where:

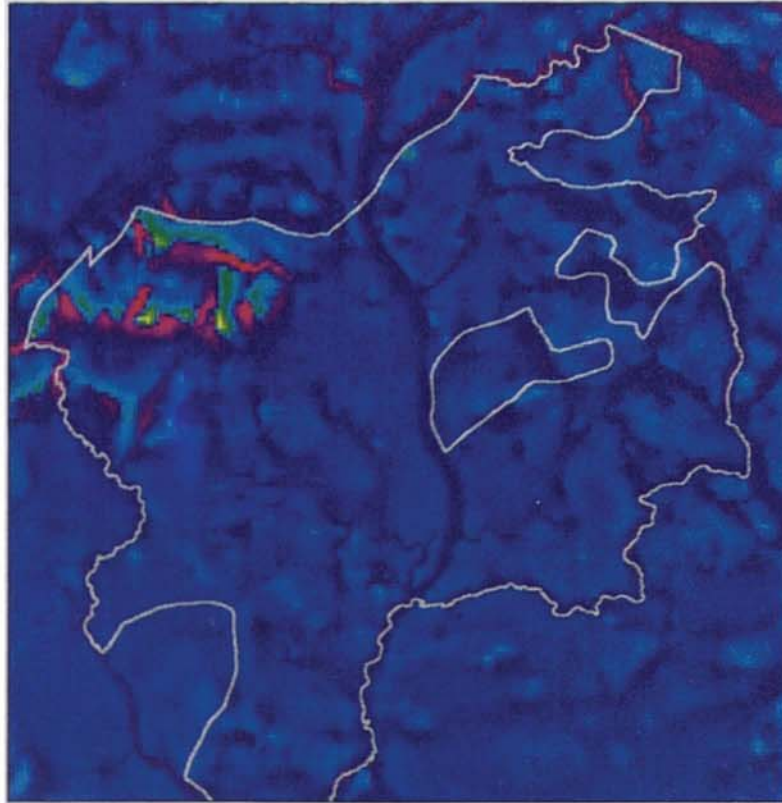


Figure 5.37: The Landscape Position Index (LPOS) raster. The scale goes from yellow, through green, cyan, blue and magenta to red. Thus red indicates the bottom of a steep valley and yellow the top of a steep hill. The blue and cyan indicate the flat, slightly rolling nature of most of the Tern area.

- LPOS* landscape position index;
 E_i elevation in i 'th surrounding pixel (L);
 E_0 elevation of central pixel (L);
 d_i distance between centre of i 'th surrounding pixel and central pixel;
 n — number of surrounding pixels.

LPOS was generated using the especially written interface to *r.mapcalc* called *r.mc.kernel*, which allows very large kernel algorithms to be quickly generated. Figure 5.37 shows the raster generated in order to get the LPOS values at the watertable observation sites. The structure of valleys and hills stands out. This LPOS raster was generated using a 1km^2 kernel with a resolution of 50m .

- Slope length length of the up slope area (m) contributing to the pixel, generated using the GRASS command *r.watershed*.
- Slope angle of slope of the pixel generated by *r.watershed*.

variable	height March	depth March	height September	depth September
x	** -0.2359	** -0.2373	* -0.2168	** -0.2831
y	* 0.1855	** 0.2463	* -0.2349	* 0.2229
DEM	*** 0.8118	*** 0.8901	*** 0.8264	*** 0.8980
LPOS	-0.0584	0.1429	*** -0.0243	0.1523
Distance	*** 0.7389	*** 0.5397	*** 0.7431	*** 0.5388
Slope length	-0.0008	** 0.2384	0.0155	* 0.2183
Slope	-0.0082	* 0.2209	0.0110	* 0.1988

Table 5.7: Pearson statistics of correlation between watertable height and depth and explanatory variables.

*** significant at 99.999%,

** 99.99%,

* 99.95%.

Since each variable is generated from the DEM, they are clearly partially dependent. It is therefore important to balance the aim of explaining as much of the variation in the dependent variable as possible, with keeping the number of explanatory variables within the resulting expression to a minimum. The inclusion of these derived variables is justifiable on the grounds that they serve to extract and integrate, into the pixel where the watertable observation has been made, the information regarding the relative heights of surrounding pixels — information which would otherwise be lost if the spot height was used alone. The east and north co-ordinates (x and y) of each point were also included as explanatory variables in the analysis. Several correlation statistics were evaluated and Table 5.7 shows the Pearson statistic for watertable height and depth in March and September 1990.

In Table 5.7 the variables east and north (x and y) show a little correlation with both watertable depth and height. Performing a linear multiple-regression on these two variables gives:

$$Wt_{mar} = 105.4753 - 0.0006506x + 0.000605y \quad (5.11)$$

$$r = 0.3384 \quad \text{significance} = -.0007$$

$$Dep_{mar} = 5.7505 + 0.001062y - 0.0009562x \quad (5.12)$$

$$r = 0.3867 \quad \text{significance} = 0.0001$$

where:

Wt_{mar} — watertable height in March 1990 (m);

Dep_{mar} — watertable depth in March 1990 (m).

Both equations (5.11 and 5.12) are of little use for predicting watertable height or depth. However, Equation 5.11 does indicate a general trend of watertable gradient from north-west to south-east, which is in agreement with the general gradient identified in Section 5.5.3. It would

be possible to fit higher order terms of x and y but this would be an exercise in curve fitting rather than reflecting any physical meaning. At a larger scale, if some satisfactory way could be found to divide the flow domain into regions of reasonably constant gradient, possibly on the basis of the *half-basin* output that can be generated by *r.watershed*, then by producing separate regression relations for each of these facets it might be possible to use x and y as predicting variables. However, a surface produced by such a method is likely to be discontinuous at the boundaries of regions unless some other smoothing function were applied. For the moment x and y are, unsurprisingly, abandoned.

The remaining variables are displayed as scatterplots against watertable height in Figure 5.38 and watertable depth in Figure 5.39. Scatterplots are generally better ways to assess correlation than statistics. The scatterplots indicate that the most correlated variables are DEM and distance from river. This relationship was elucidated in the previous section (Section 5.6.3). It is a common feature of unconfined watertables that they tend to follow a form that is a subdued version of the land surface topography, Figure 5.40.a. In other catchments where there is a considerable land surface height variation this can be partly explained by more rainfall on the higher ground. However the land surface height difference in the Tern region is not great, certainly not great enough to cause very large variations in rainfall so the variation of watertable can again be put down to drainage to the river system, removing water from the low-lying areas.

Figures 5.38.c and 5.39.c indicate that LPOS is not well correlated, however this is only an preliminary investigation and there is considerable scope for adjusting LPOS. These adjustments that might increase its power as an explanatory variable because by decreasing the size of the kernel larger scale features (e.g. smaller river valleys) are picked out and smaller scale structure is lost. It is likely that LPOS needs some fine tuning to pick out the scale of landscape structure that the watertable responds to (i.e. Figure 5.40).

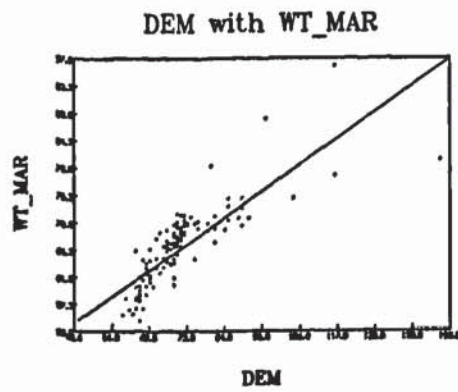
Slope and slope length were included because they are simple to generate. It is possible that they correlate more closely to watertable gradient than to absolute height or depth, this requires further research.

Multiple linear regression using the most correlated variables, DEM and distance, gives the following relationships:

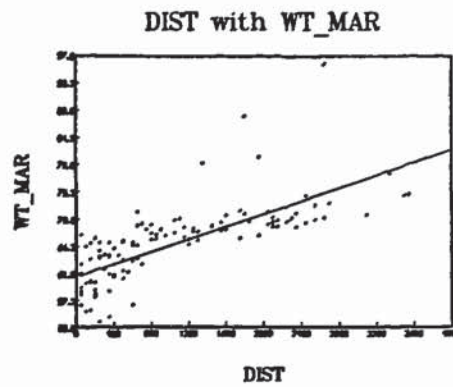
$$\begin{aligned} Wt_{mar} &= 44.2734 + 0.02819 \times DEM + 0.002338 \times DIST \\ r &= 0.8444 \quad \text{significance} < 0.0001 \end{aligned} \quad (5.13)$$

$$\begin{aligned} Dep_{mar} &= -42.0828 + 0.6892 \times DEM - 0.001794 \times DIST \\ r &= 0.8993 \quad \text{significance} < 0.0001 \end{aligned} \quad (5.14)$$

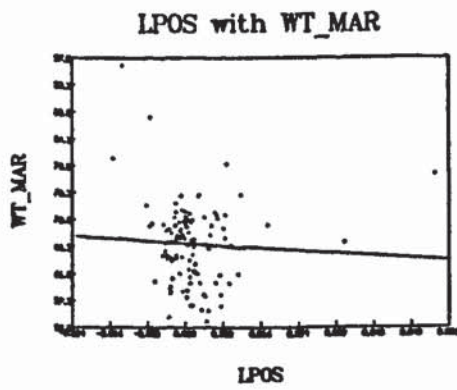
$$Wt_{sep} = 43.3731 + 0.2902 \times DEM + 0.002236 \times DIST \quad (5.15)$$



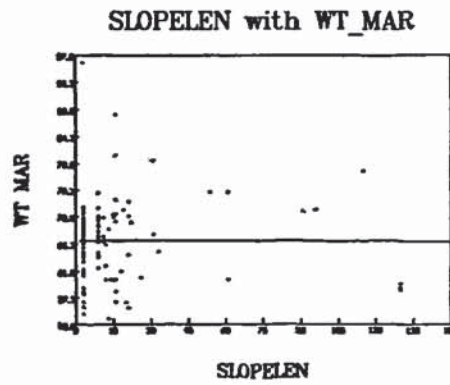
(a)



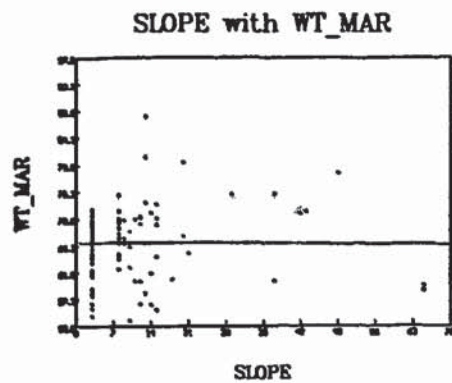
(b)



(c)

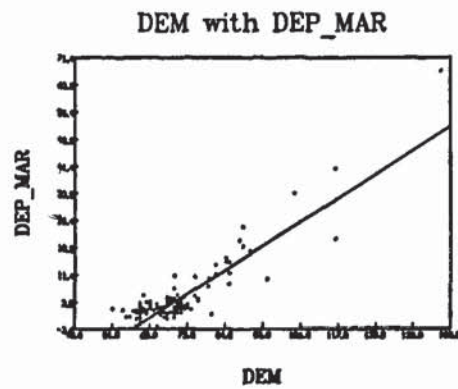


(d)

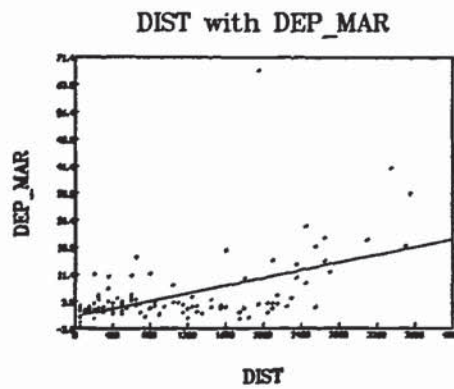


(e)

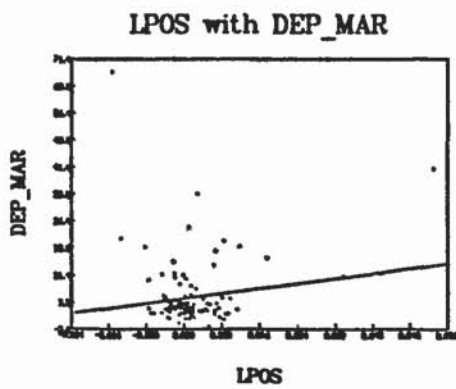
Figure 5.38: Graphs of watertable height plotted against (a) land-surface height (DEM), (b) distance from river (DIST), (c) landscape position index (LPOS), (d) slope length (SLOPELEN) and (e) slope gradient (SLOPE).



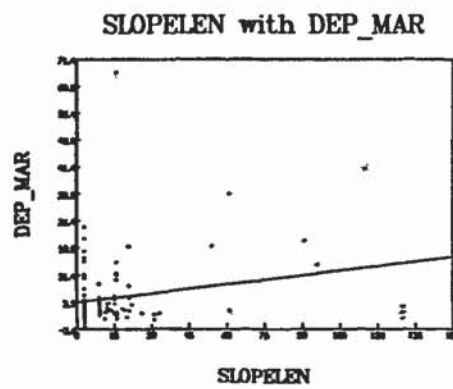
(a)



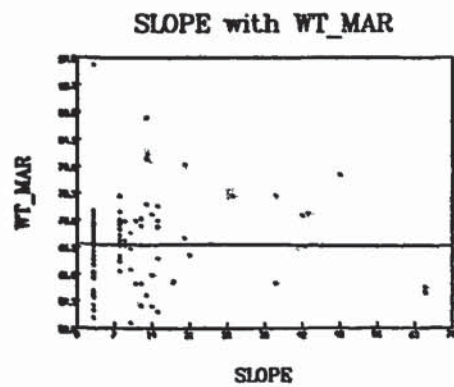
(b)



(c)



(d)



(e)

Figure 5.39: Graphs of watertable depth (WT_MAR) plotted against (a) land-surface height (DEM), (b) distance from river (DIST), (c) landscape position index (LPOS), (d) slope length (SLOPELEN) and (e) slope gradient (SLOPE).

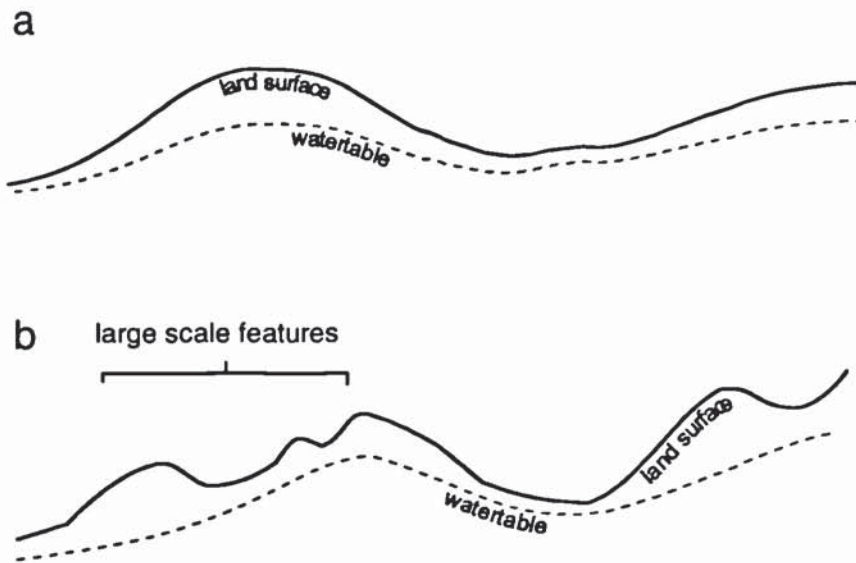


Figure 5.40: Variation of watertable with land surface height. (a) the general relationship between land surface and watertable and (b) some structures scales may not be reflected in the watertable surface.

$$r = 0.8562 \quad \text{significance} < 0.0001$$

$$Dep_{sep} = -41.4463 + 0.6891 \times DEM - 0.001897 \times DIST \quad (5.16)$$

$$r = 0.9084 \quad \text{significance} < 0.0001$$

The advantages of using a regression equation are that once it has been generated then it is simple to express in a map algebra language, e.g in *r.mapcalc* notation Equation 5.13 is:

r mapcalc "wt_surface=-44.2734 + 0.02818 * dem + 0.002338 * distance_from_river"

to fully realise Clarke's (1994) statistical model (Equation 5.9) a random variable representing the possible variation around the mean (given by this expression) needs to be added. One main disadvantage is that where a real observation exists the generated watertable surface will probably not match it.

Because of all the disadvantages of attempting to specify a hydraulic head surface for a ground-water model discussed in Section 5.9 this evaluation was not taken any further.

5.7 Geostatistical investigation of the watertable surface

The position of the watertable is one of the most important factors in accessing the vulnerability of crops to groundwater abstraction. The watertable reading dataset from the NRA represents a valuable resource for specifying the position. This can be used for cartographic techniques such as Hedges's (1989) model Chapter 4, as an input for a groundwater model such as Modflow or as an alternative to using a groundwater model (one might, for example, generate a sequence of surfaces from the data that represented a typical year). The problem is how to generate a continuous surface from what is essentially a set of point observations in space and time. In Section 5.5.2 this problem was tackled by using two interpolation algorithms implemented within GRASS. In this section the utility of geostatistical algorithms realising the value of this dataset is assessed.

In the previous section (Section 5.6) the watertable data were shown to exhibit some spatial properties. This section will further explore these properties and make some recommendations for future research. The exploratory data analysis (EDA) performed here loosely follows the schedule suggested by Pannatier (1994).

5.7.1 What is geostatistics?

Geostatistics is a branch of statistics concerned principally with properties that vary and are correlated spatially. It was developed by mining engineers for evaluating ore yields, though some of the ideas have developed in parallel within other disciplines under other names. Recently geostatistical techniques have been applied to many other phenomenon.

Geostatistics is a very complex subject and only a brief review of the major principals is given here, the reader is referred to Cressie (1993), Isaaks and Srivastava (1989) and Clark (1979) (in ascending order of comprehensibility and descending order of complexity) for further reading.

When examining the relationship between variables it is common practice to plot the data as a scatterplot (such as those in Figures 5.38 and 5.39 above) and to give some measure of their association in the form of a statistic of correlation (there are several correlation statistics in the literature). It is a basic paradigm in geostatistics that a variable measured at a point in space will show some correlation with the same variable measured at some other point, and that this correlation will weaken the further apart the two points are. As in classical statistics, association can be visualised as a scatterplot, the scatterplot in this case is made up of pairs of points separated by a distance commonly referred to as lag or h^6 — giving the term *h*-scatterplot. Figure 5.41 a, b, c and d are *h*-scatterplots at different lags.

It becomes cumbersome dealing with many *h*-scatterplots for different lags and so the data is summarised by plotting some measure of correlation against the lag. The classical correlation coefficient or covariance are often plotted, but the measure most frequently used by geostatisticians is the semivariogram. The semivariogram is half the average squared difference between the paired

⁶ Standard mathematical vector notation is used to indicate that lag is a distance and a direction.

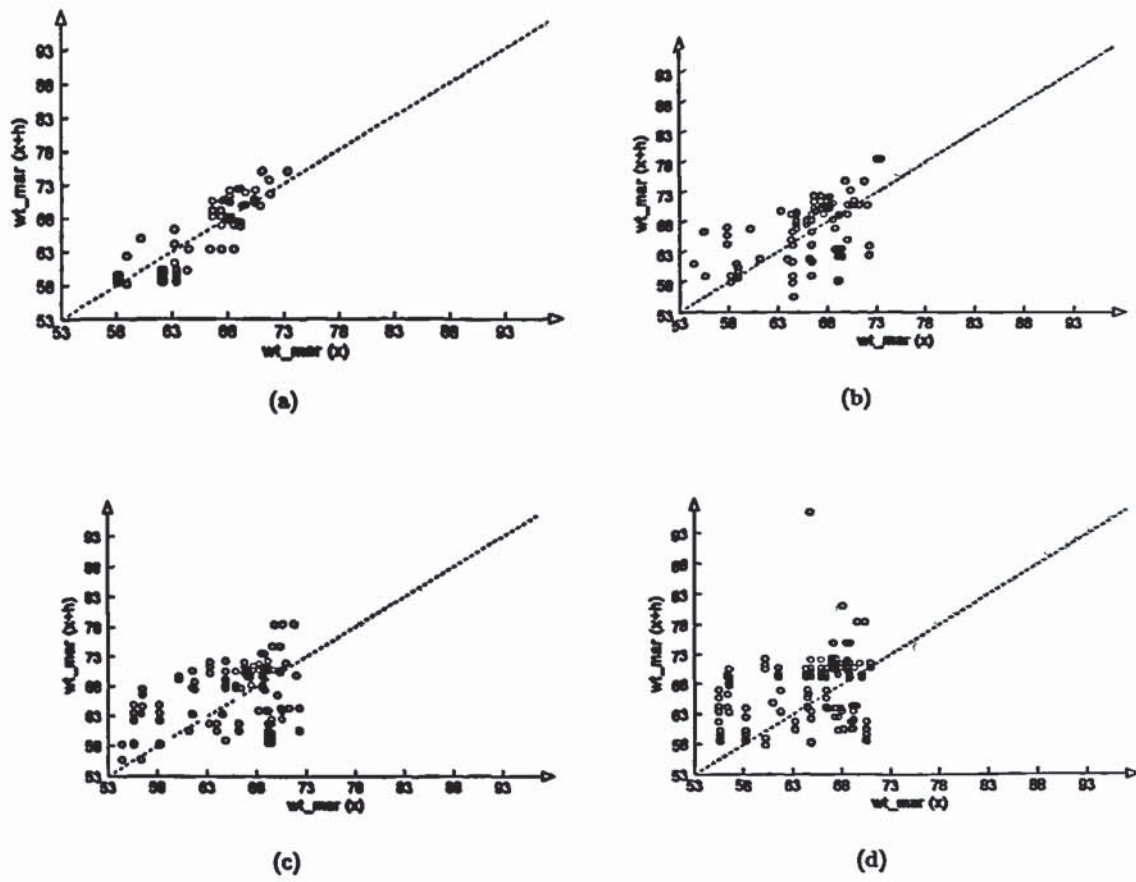


Figure 5.41: Examples of h-scatterplots at increasing lags. (a) 1000m, (b) 2000m, (c) 3000m and (d) 4000m. In this case the variable is watertable height in March 1990.

data values (Equation 5.17).

$$\gamma(h) = \frac{1}{2n(h)} \sum_{i=0}^n (v_i - u_i)^2 \quad (5.17)$$

where:

$\gamma(h)$ — semivariogram measure of correlation, over a lag of h .

h — lag distance separating paired data points (L);

n — number of data point pairs;

v_i — first point in i 'th data pair, separated from u_i by vector $-h$;

u_i — second point in i 'th data pair, separated from v_i by vector h .

The semivariogram gives an indication of the spread from the 45° line. This is useful because h-scatterplot have the property that at total correlation, i.e. when the lag is zero or each observation is plotted against itself, the plotted points fall on the 45° line. The literature can be confusing and the reader should be aware that the term semivariogram can be used in three associated but slightly different ways.

- The *real* semivariogram is how the correlation decays in reality. This is never completely accurately known.
- The *experimental* semivariogram is the plot derived from the observed data, and gives an indicate of the form of the real semivariogram.
- The *model* semivariogram is fitted to the experimental semivariogram. This model attempts to match a mathematical function, possibly linear, but more often spherical or exponential or some combination, to the experimental semivariogram.

Many authors have suggested alternatives to the semivariogram, which share similar properties to the classic semivariogram described above but are usually more robust (less affected by aberrant data) or have some other desirable property. However, these are beyond the scope of discussion here.

There are some common terms in geostatistics:

- Lag — the distance between the paired data points in the h-scatterplot or semivariogram.
- Support — the size of the *block* from which a data is sampled. Commonly data is assumed to have been sampled at a point, but this is frequently not the case. Large supports generally reduce variance.
- Range — the distance beyond which points are no longer correlated and the semivariogram plot levels off.

- Sill — a common feature of semivariogram plots, it is the maximum value that the semivariogram reaches at some distance (the range) beyond which the two data points of a pair are effectively uncorrelated.
- Tolerance — when gathering data pairs semivariograms, it is common practice to allow some tolerance on both the direction and magnitude of the lag, thus pairs can be grouped and the semivariogram will not be so jagged.
- Variogram — generally used carelessly to mean semivariogram — no distinction when using the terms variogram and semivariogram is intended in this project.

The principal application of the geostatistical theory reviewed so far is in the prediction of an attribute at an unobserved point from scatter observations. The principal tool for applying this theory is the process known as kriging⁷. Kriging is a spatial estimation or interpolation algorithm. The inverse distance and spline interpolators used in Section 5.5.2, are also examples of spatial estimators. They both estimate the value of a variable at a point by using local observations but assume very different underlying models of spatial distribution. What they have in common is that the underlying model, whether distance weight average or spline, is largely arbitrary and may have little in common with the properties of the real spatial variation. The aim of fitting a semivariogram model to the experimental semivariogram is to allow the kriging estimator to use the way correlation actually decays with distance and so improve the estimation.

Isaaks and Srivastava (1989) defined several criteria for a good estimator.

1. It should produce an estimation with a distribution with statistical properties (mean, variance etc.) similar to the true distribution.
2. The distribution of the residuals (the error or difference between the estimated and the real value) should be unbiased; i.e. the interpolator should not consistently over or underestimate. Thus the residual distribution will have a mean of 0.
3. The distribution of residuals should not be too skewed; i.e. median and mode of the residual distribution should also be near zero.
4. The distribution of residuals should not be too spread; i.e. it should have low variance.
5. The residual distribution should exhibit unconditional bias; that is if the distribution is divided into smaller ranges these also should be unbiased.

Isaaks and Srivastava accepted that there will inevitably be a trade off amongst these criteria. It is seldom possible to check how well an estimator performs against these criteria unless a second set of data is available.

⁷ Actually kriging is a family of techniques, but only ordinary and universal kriging, and cokriging will be addressed here.

Most estimators satisfy the unbiased criteria (number 2) and the geostatistical estimation technique kriging is no exception. But, according to Isaaks and Srivastava, kriging is also the "best linear unbiased estimator" i.e. it goes further in satisfying criteria 4 than any other estimator.

The variance of the estimate residuals is kept to a minimum by assuming that the 1 to n observed values and the value to be estimated are random variables that conform to a stationary random function model. Each unobserved point will therefore have the same expected value, and for any pair of points the joint distribution depends only on the displacement between the points and not on their absolute locations. As with the inverse distance model, kriging uses a weighted average, the challenge is to find a set of weights that minimise the variance without compromising the property of no bias.

$$\hat{V}(x_0) = \sum_{i=1}^n w_i V(x_i) \quad (5.18)$$

where:

- $\hat{V}(x_0)$ — the estimate of the random variable at location x ;
- $V(x)$ — the true random variable at location x ;
- x_i — location of i 'th random variable;
- w_i — weighting of the i 'th random variable.

The unknown in Equation 5.18 is the set of weights w_i , these can be generated from the following equation (the derivation is given in Isaaks and Srivastava, 1989):

$$\mathbf{w} = \mathbf{C}^{-1} \times \mathbf{D} \quad (5.19)$$

where:

- \mathbf{w} — an $(n+1) \times 1$ matrix of weights;
- \mathbf{D} — an $(n+1) \times 1$ covariance matrix between $\hat{V}(x_0)$ and $V(x_1), \dots, V(x_n)$;
- \mathbf{C} — an $(n+1) \times (n+1)$ covariance matrix between $V(x_1), \dots, V(x_n)$ and $V(x_1), \dots, V(x_n)$;

or

$$\begin{bmatrix} w_1 \\ \vdots \\ w_n \\ \mu \end{bmatrix} = \begin{bmatrix} \tilde{C}_{11} & \dots & \tilde{C}_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{C}_{n1} & \dots & \tilde{C}_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \tilde{C}_{10} \\ \vdots \\ \tilde{C}_{n0} \\ 1 \end{bmatrix} \quad (5.20)$$

where:

- \tilde{C}_{ij} — covariance between i 'th and j 'th random variables;
- w_i — weight of the i 'th random variable;

μ — Lagrange parameter, added as part of the solution.

As the covariance matrix is generated from the model semivariogram⁸, one of the most vital task is fitting the most appropriate model to the experimental semivariogram.

Universal kriging might be better termed “kriging with a prior trend model” (Deutsch and Journel, 1992). Basically the underlying random variable model is partitioned into a trend component and a residual. It is preferable for the trend component to be based upon the physics of the process. An extension of universal kriging is “kriging with an external drift” (Deutsch and Journel, 1992). Here a secondary variable is used to describe the drift or trend. For estimating the water table land surface height might proved a suitable external drift.

The use of further variables leads into the final subject that will be tackled in this very brief review of geostatistics; that is the cross-h-scatterplot and the method of cokriging. A cross-h-scatterplot is similar to an h-scatterplot only a dependant variable is paired with an independent variable at a given lag. Thus when $|h| = 0$ this will be a normal scatterplot.

Ordinary cokriging is similar to ordinary kriging accept that measures of cross-correlation with other variables — variables which may have better sampling properties — are included in the method (Equation 5.21). In practice cokriging is seldom used, rather universal kriging and kriging with a drift are generally preferred (Deutsch and Journel, 1992).

$$\hat{u}_0 = \sum_{i=1}^n a_i \cdot u_i + \sum_{j=1}^m b_j \cdot v_j \quad (5.21)$$

where:

- \hat{u}_0 — estimate of variable at location 0;
- u_i — observation of variable at location i ;
- a_i — weighting of observation u_i ;
- v_j — observation of secondary variable at location j ;
- b_j — weighting of observation v_j .

Variance is minimised in the same way as for kriging (Isaaks and Srivastava, 1989).

5.7.2 Geostatistics of the watertable

For this investigation the same set of data was used as for the statistical analysis (Section 5.6.3). Pannatier (1994) suggested a sequence of operations for performing exploratory data analysis (EDA) of a dataset. The first stages are the univariate and multivariate analysis, which have already been performed in Sections 5.6.1 and 5.6.3. In recent years the process of EDA has moved from the systematic generation of telling statistics to a much more graphical exploration of the

⁸ The process to generate a covariance matrix from the model semivariogram is beyond the scope of this review, see Cressie (1993).

data. This is a result of the improvement in computer power, statistical software packages and graphical display hardware. The subsequent stages of Pannatier's (1994) schedule were performed with the VARIOWIN package (Pannatier, 1994), and are presented below.

Anisotropy

A variogram surface is a way of detecting anisotropic effects, it is a two dimensional graphical plot of the semivariogram. Figure 5.42 shows variogram surfaces for watertable heights and depths for March 1990. There appears to be anisotropy in both plots. In the watertable height plot (Figure 5.42.a) the correlation appears to decay less slowly in the north-south direction, whilst in the depth plot (Figure 5.42.b) the direction is tending towards north-west to south-east. An examination semivariogram plots in these directions confirms this interpretation (Figure 5.44 a, b, c & d).

One would expect correlation to be stronger in a direction normal to the groundwater gradient, particularly for watertable heights, as naturally a displacement in the direction of gradient involves a change in height whereas a direction normal to the gradient does not (Figure 5.46). So this may give an indication of the major flow directions in the groundwater system, as identified in Section 5.5.3. Figure 5.43 gives the variogram surface of a subregion (shown in Figure 5.47) where the gradient might be expected to be more consistently in one direction and Figure 5.45 show the semivariogram plots. The anisotropic pattern does indeed appear stronger, although the reduction in the number of samples makes this less certain.

Cross-correlation

It was shown in Section 5.6.3 that watertable height and depth were strongly correlated with both elevation and with distance from rivers. Both these variables have the property of good spatial coverage of observations. Figure 5.48 gives variogram surfaces for watertable height and depth with elevation and distance from river. Again there is anisotropy and again the direction of anisotropy associated with the depth variable appears slightly offset with respect to height. Figure 5.49 shows the effect of this anisotropy on directional semivariogram plots. These relationships mean these secondary variables could be used in a *kriging with an external drift* or cokriging approach.

An interesting feature is that the east-west semivariogram plots tend to dip after the sill has been reached. This may just be an artefact of the low number of observed measurement pairs at these long lags or it may indicate some kind of cyclic structure of the landscape. Landscape forming processes, such erosion, deposition, soil creep and river network formation, can give a landscape a rolling nature. The hill-valley structure of a landscape may be regular enough that it can almost be considered to have a wavelength in a given direction. Whether this is the case in Shropshire and whether this accounts for the apparent cyclic feature in the east-west semivariogram can only be determined with a larger dataset covering a greater region.

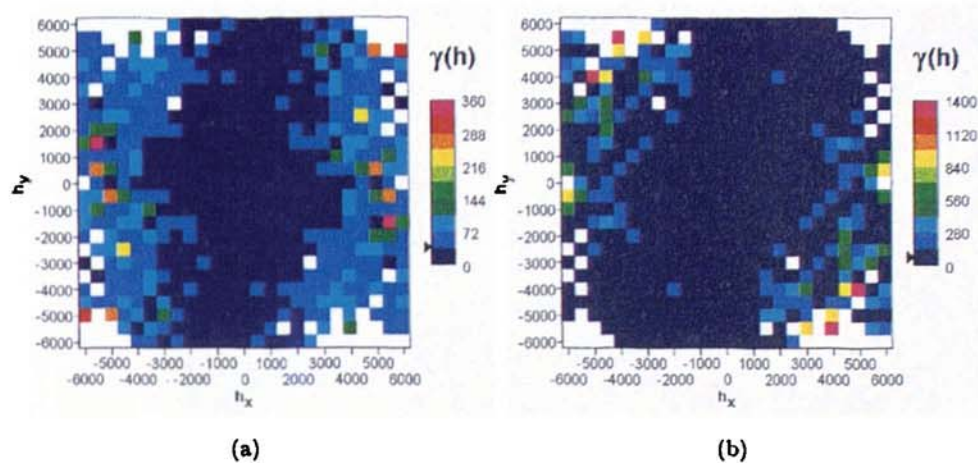


Figure 5.42: Variogram surfaces for watertable (a) height and (b) depth, March 1990. The arrow against the legend indicates the variance of the whole dataset.

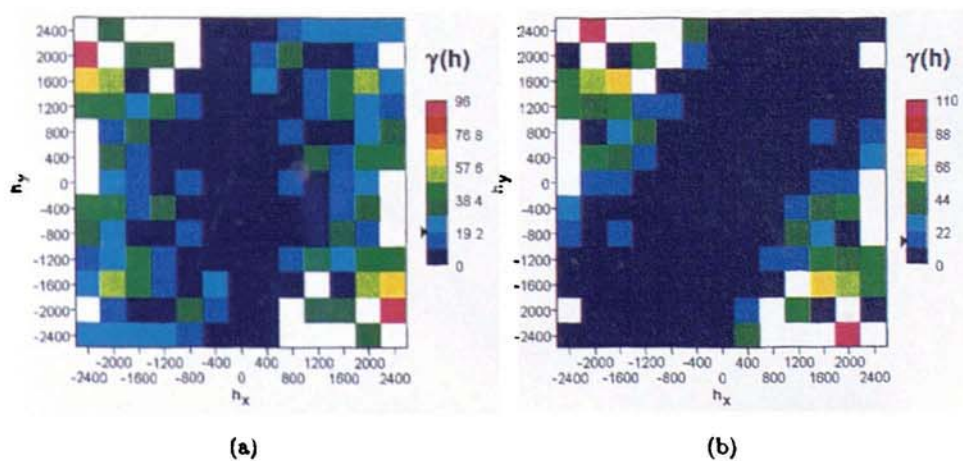


Figure 5.43: Variogram surfaces for watertable (a) height and (b) depth, in March 1990, for region 360000-365000E and 323000-328000N.

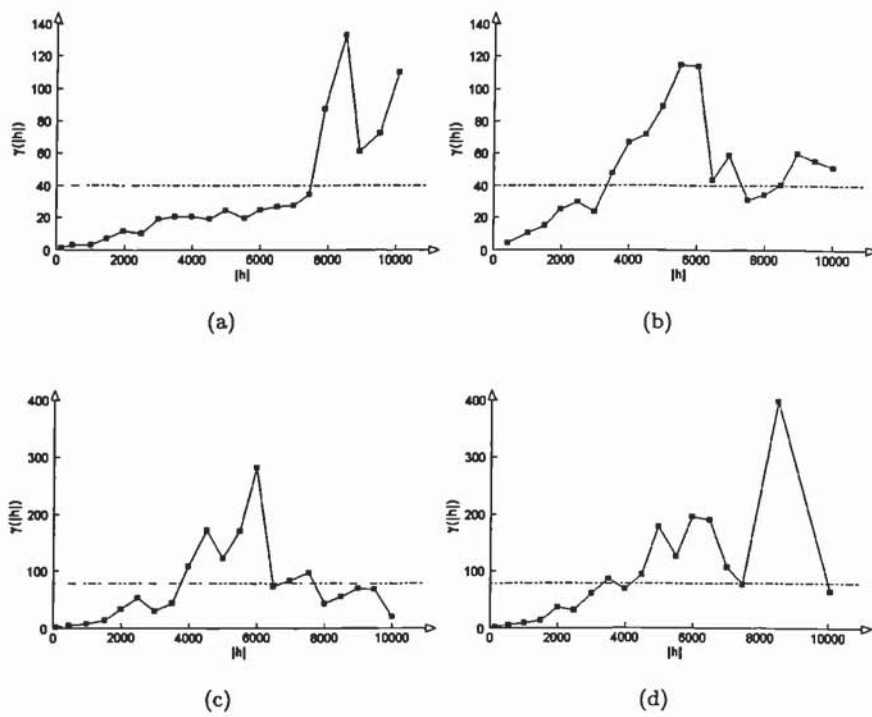


Figure 5.44: Semivariogram plots for watertable height in (a) north-south and (b) east-west directions and watertable depth in (c) 20°E and (d) 110° E directions. Using a lag of 1000m and a lag tolerance of 500m, angular tolerance 20°. The dotted line indicates the whole dataset variance.

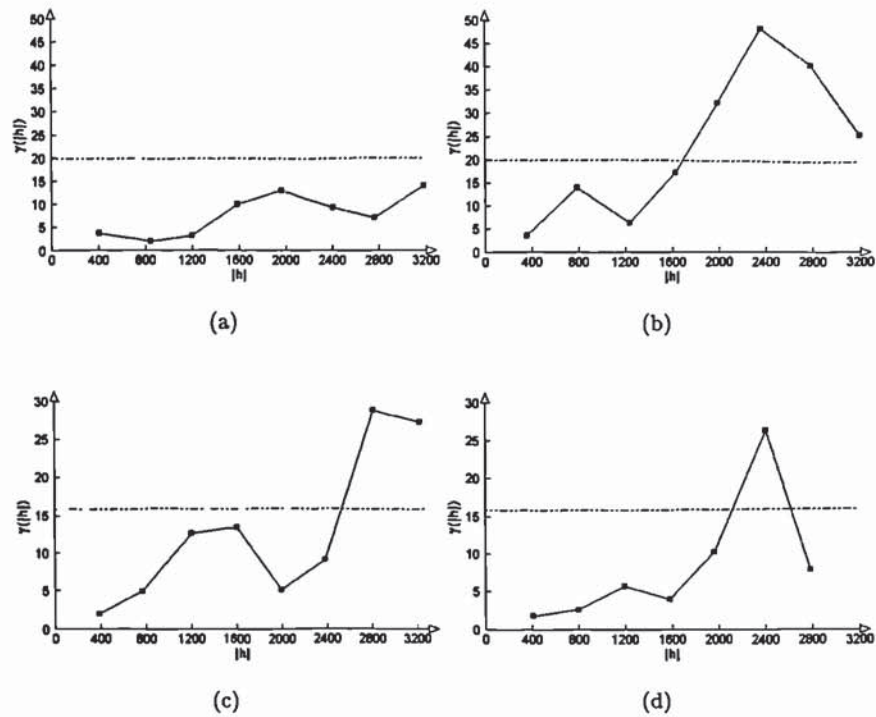


Figure 5.45: Large scale semivariogram plots for watertable height in (a) north-south and (b) east-west directions and watertable depth in (c) 20°E and (d) 110° E directions. Using a lag of 400m and a lag tolerance of 200m, angular tolerance 20°. For region 360000-365000E and 323000-328000N

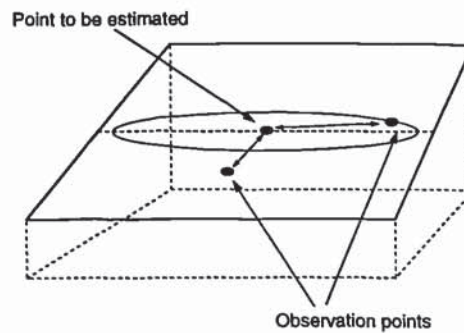


Figure 5.46: A simple geostatistical estimation situation. Here the height of a point on an unknown plane must be estimated from the heights of two distant measured points. Naturally the correlation is stronger in the direction normal to the gradient, thus the observed measurement which is further from the point to be estimated should be given more weight than the closer point. The ellipse gives an indication of the range of the model semivariogram which might be chosen if the nature of the surface were not known and there were only a few observation points.

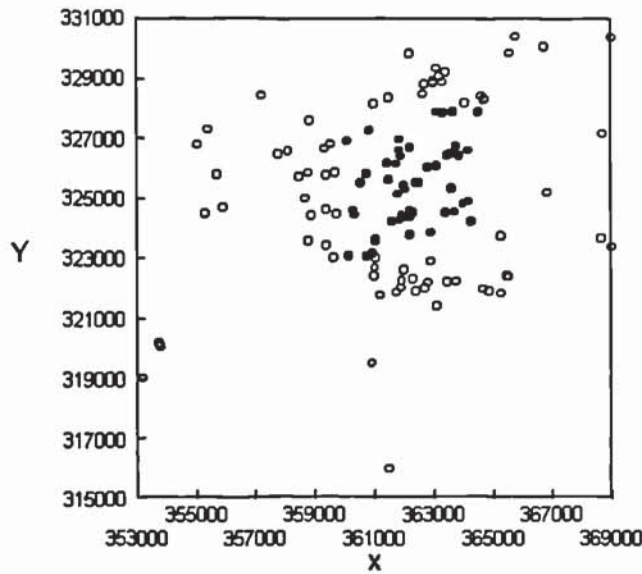


Figure 5.47: Geostatistical modelling region and subregion showing location of the observation wells. The circles and dots indicate the locations of the observations, the dots are those also used in the subregion analysis.

Semivariogram models

Figure 5.50 shows two standard models fitted to the experimental semivariograms. Neither is satisfactory and it is likely that a combination of models would prove better. The selection of models is largely arbitrary, it is an art as much as a science and owes a lot to experience.

5.7.3 Evaluation

In Section 5.5.2 watertable surfaces were interpolated using the facilities available within the GRASS environment; i.e. a distance weighted average (*r.surf.idw*) and a spline based interpolator (*r.surf.tps*). After some manipulation and investigation the distance weighted average function was selected as the standard way of generating watertable surfaces for reasons that were as much practical as theoretical. From the analysis in this section it is clear that the weightings assigned to the observation on the basis of inverse distance are not optimal in that they do not reduce the variance of error to a minimum. However, by using the GIS's internal interpolator, other knowledge such as location of the rivers could be easily added (in a practical sense) to the interpolation and given that the accurate specification of a initial hydraulic head is not necessary and in fact not desirable (see Section 5.9 for a discussion of this) the inverse linear weighting was considered acceptable.

Section 5.7 has described a brief investigation into the geostatistical properties of the watertable observation dataset. There is clearly a great deal of scope for improving spatial esti-

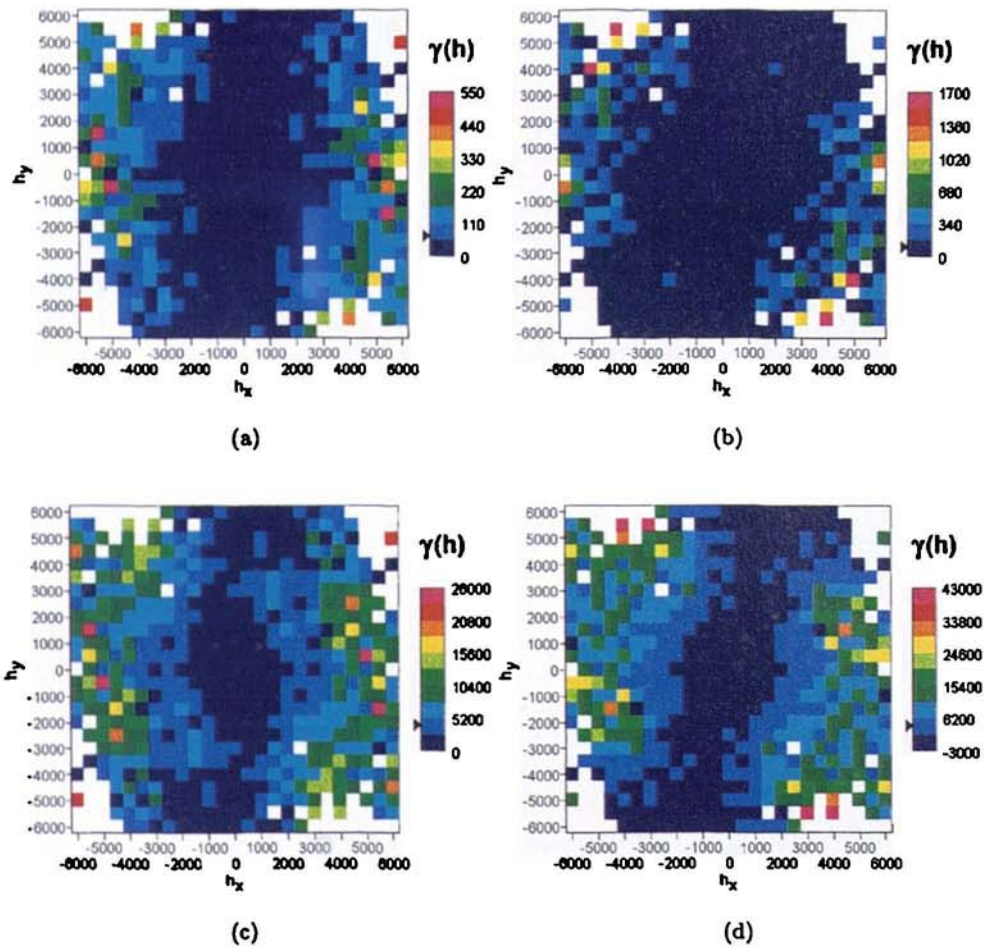


Figure 5.48: Cross-semivariogram surfaces for (a) watertable height and elevation, (b) watertable height and distance from river, (c) watertable depth and elevation and (d) watertable depth and distance from river. The arrow against the keys indicate the covariance of the dataset.

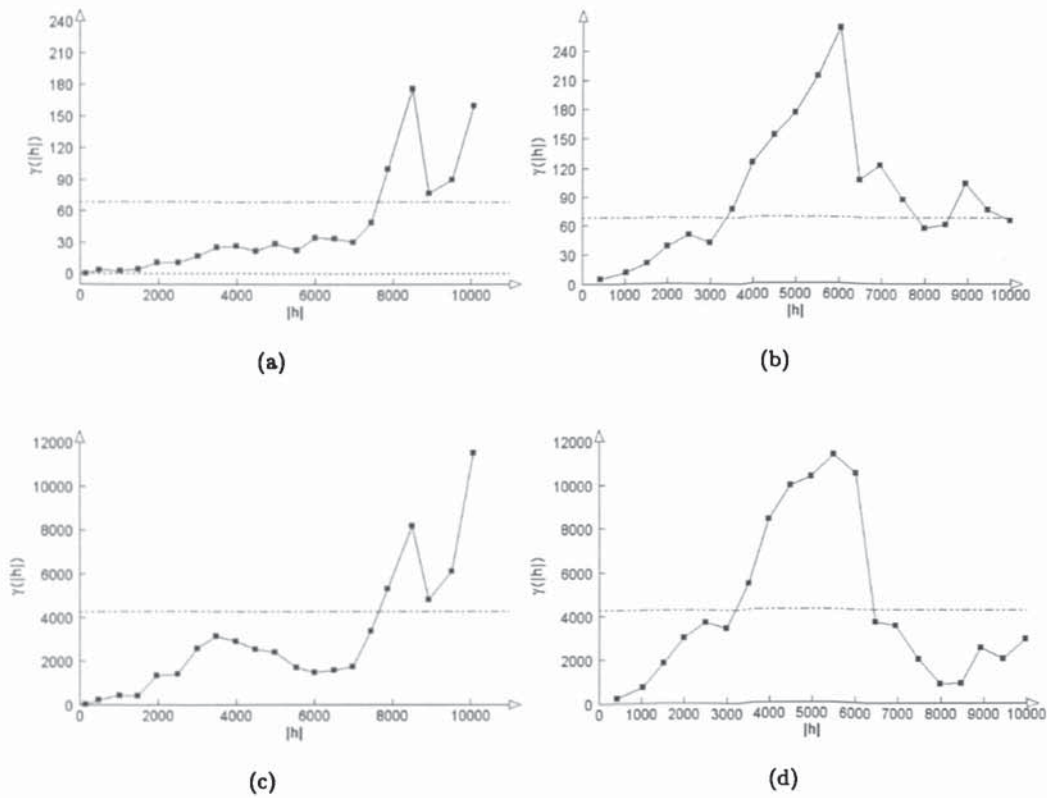
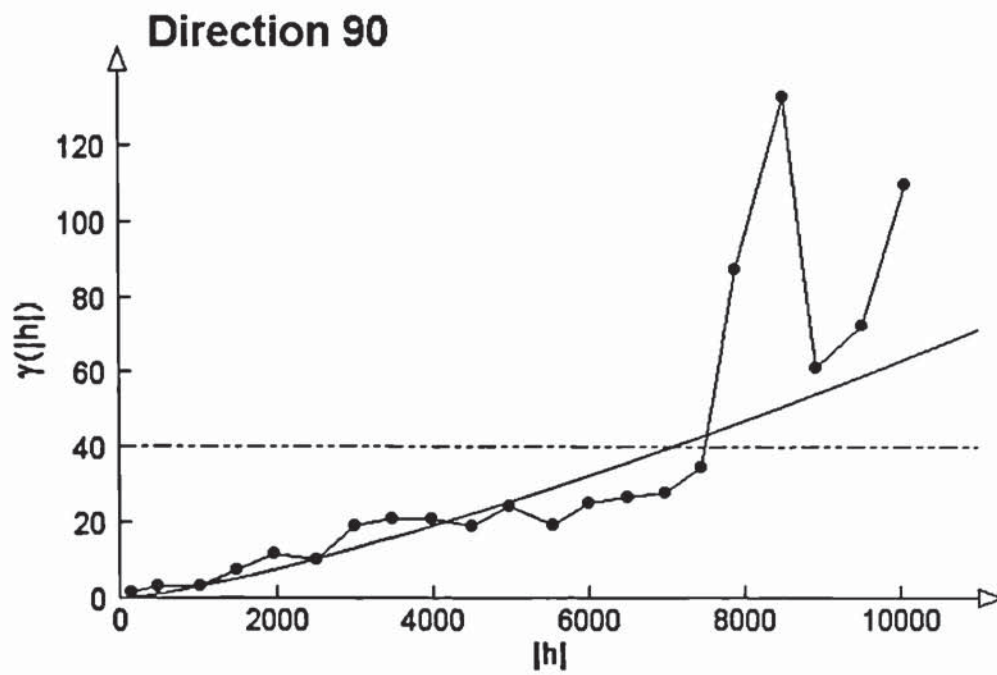
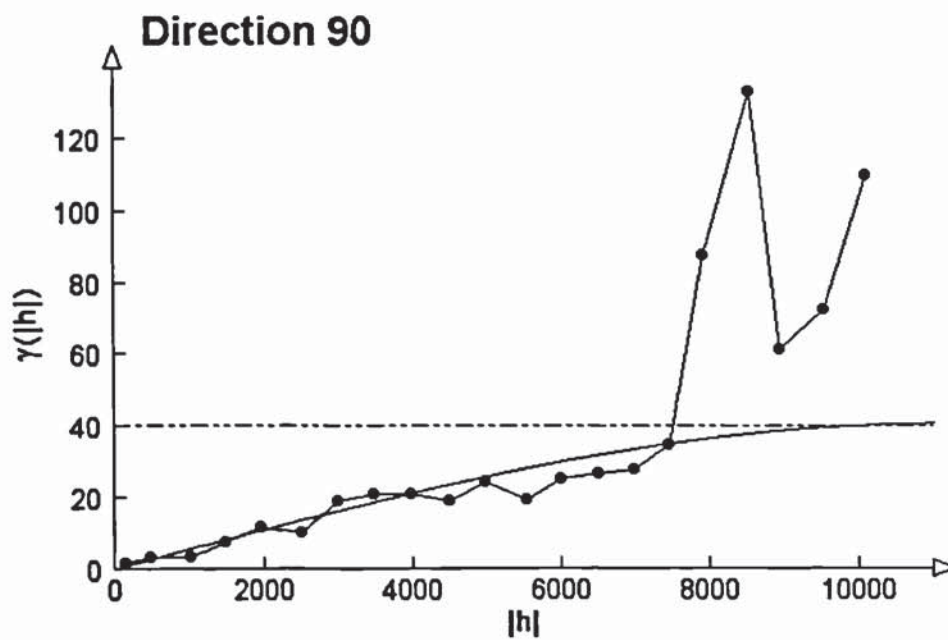


Figure 5.49: Cross-semivariogram plots for watertable height readings in (a) north-south and (b) east-west directions and watertable depth readings in (c) 20°E and (d) 110° E directions. Using a lag of 1000m and a lag tolerance of 500m, angular tolerance 20°. For region 360000-365000E and 323000-328000N. The dotted line indicates the covariance of the dataset



(a)



(b)

Figure 5.50: (a) a linear model and (b) a spherical model fitted to the semivariograms of watertable height. The dotted line indicates the whole dataset variance.

mation of watertable, beyond that of the interpolators available within GRASS, using kriging and including secondary variables and accounting for anisotropy. Also, though it has not been investigated here, one can see that the groundwater surface can be expected to be temporally as well as spatially auto-correlated, and it is recommended that this feature is included in a future investigation.

Currently GRASS is not linked to a comprehensive geostatistical package, consequently there was considerable work involved in moving the data to and from GRASS on the UNIX machines to the statistical and geostatistical packages which are PC based. This was one of the practical reasons that the internal GRASS interpolator was chosen. On the subject of linking a GIS to a geostatistical package it is interesting to note that the strategy outlined in Section 3.5.1 is amenable to achieving geostatistical packages as well as process models. However, because EDA (exploratory data analysis) is an interactive process there are some distinct advantages to a more closely integrated approach advantages that do not apply when linking process models. In particular the facility of linked brushing as defined by Majure, Cressie, Cook and Symanzik (1996) could not be implemented with a loose-linked approach. Link brushing is the facility whereby when a feature, say an observation point, is highlighted in one window that highlight is repeated in all other windows that include that point. Thus when, for example, a point is identified as an outlier on a scatterplot, it would be useful if that same point when selected was also highlight in a map of the observation points. Thus any spatial reason for its anomaly may be more clearly seen, i.e. it may be located in a completely different region to the other points.

In the same way that GIS *adds value* to a process model it also adds value to a geostatistical package which clearly relies on spatial data.

5.8 Modelling the Tern area

Having built up a conceptual model of the groundwater system in Section 5.5. This section deals with some of the various exercises in using Modflow to perform numerical modelling.

5.8.1 The 1990 recession

As a first exercise the 1990 groundwater recession was modelled. There are a number of considerations to take into account here. Within the hydraulic head field generated according to Section 5.5.2 the difference in hydraulic head across the groundwater unit is approximately 80m, however, the recession of the watertable during the summer months although variable is seldom greater than 1m, Figure 5.51 (see Section 5.6.1). There is therefore considerable potential for groundwater movement in the system, and the risk is that, in calibration, transmissivity parameters will be reached that are so low that they effectively freeze the system and thus give a result very close to the observed, but not because the system was properly modelled.

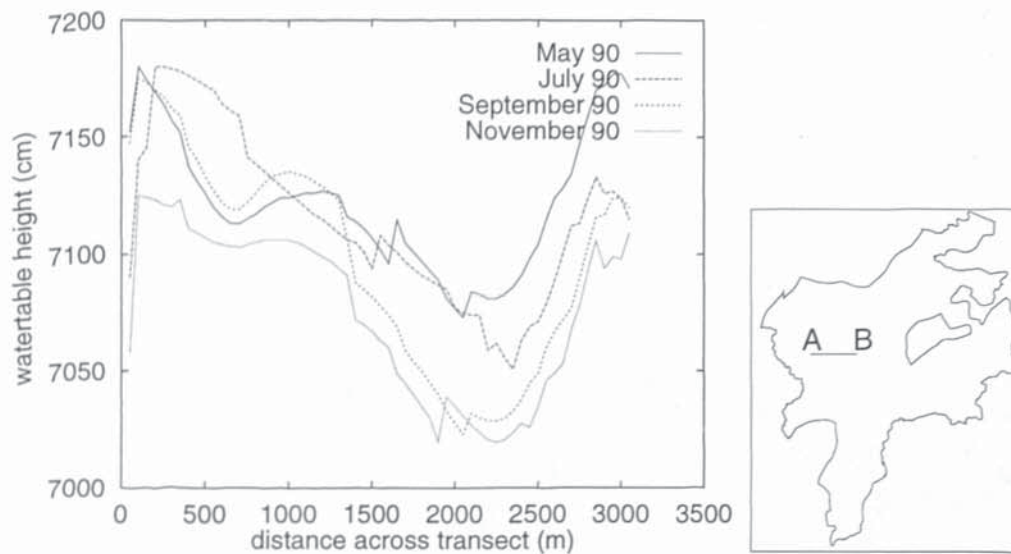


Figure 5.51: Profile of the watertable showing the 1990 recession. Profiles extend from A to B in location map.

The modelling strategy was to begin with a very simple model and add refinements as understanding of the problem was gained. The first model used a single layer with a constant transmissivity and confined and unconfined storage coefficients, but a varying initial hydraulic head and a varying aquifer top (defined by the DEM).

The MORECS model (see Section 6.1.3) for 1990 (Figure 5.52) predicts that a soil moisture deficit builds up in March and is not relieved until October so initially recharge is ignored.

Figure 5.53.a shows the interpolated November watertable generated according to Section 5.5.2, Figure 5.53.b shows the modelled watertable and Figure 5.53.c shows the difference between the two.

Certain features stand out. Firstly there is no error at the fixed-head boundaries. This is simply because both the groundwater model and the interpolated surfaces have been forced to conform to the fixed-head boundaries at these points. The second and more important feature is the rise in the watertable adjacent to the Tern seeming to represent artesian conditions. This is because of an inadequacy in the simple model, for here the conceptual model indicates that the Tern acts as a partial fixed-head boundary and this assumption was explicitly used when generating the watertable surface (Section 5.5.2).

It can be seen from Figure 5.54 the model tends towards the steady state solution for the model boundaries selected, as expected the model with the highest transmissivity tends toward this solution the fastest.

The first refinement to the model is to include the assumption of hydraulic connectivity between the river and the groundwater. The figure of $1.5 \text{ m}^3 \text{ km}^{-1} \text{ day}^{-1}$ is used for river leakage

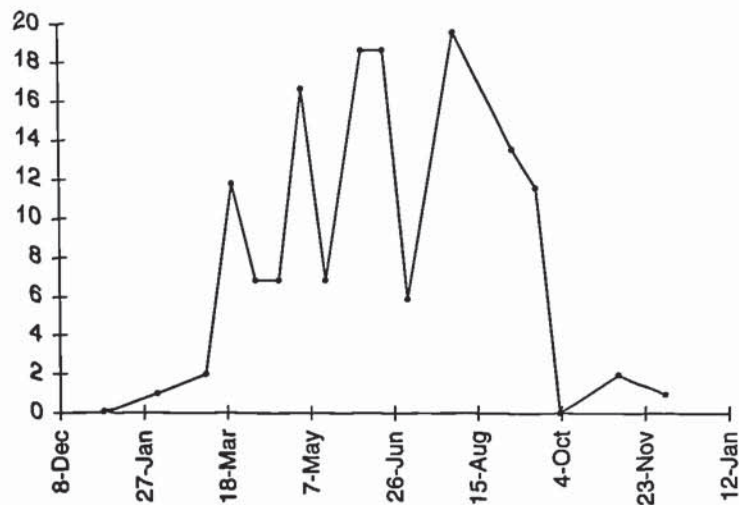


Figure 5.52: MORECS soil moisture deficit data for 1990.

(Section 5.5.7). Figure 5.55.a shows a modelled water table surface, Figure 5.55.b shows the difference between the modelled and the Figure 5.55.c shows cross-sections of the watertables for comparison. It can be seen that the model which includes river leakage as expected more closely resembles the interpolated surface near the river.

The next refinement is to add the effect of faults in the groundwater unit. In Section 5.8.2 and Section 5.5.9 the effect of faults in the region is discussed and modelled. It was found that, for the most part, the faults of the region act to restrain flow. For this model instead of having 1 single transmissivity value for the unit and lower transmissivity value will be assumed for elements in fault zones. A raster representing transmissivity with faults was generated using *r.mapcalc*.

```
r.mapcalc "transmissivity=if(isnull(fault),400,50)"
```

Surprisingly there is little difference in the result. Figure 5.56.a shows the difference between the models with and without low flow faults, Figure 5.56.b shows the difference between the interpolated surface and the surface modelled with faults and, Figure 5.56.c shows profiles of the modelled watertable under different assumptions. No further additions or complication improved the model any further.

5.8.2 The 1984 test pumping

1984 saw the commissioning of the Tern 1 group of boreholes. It was also the first time the scheme was used for river regulation in a drought. Consequently there is a lot of activity and movement in the groundwater system.

As a test of the Modflow-GRASS link some for the pump tests were modelled, these provided useful information on the behaviour of the aquifer as well as evaluating the link. Given below are

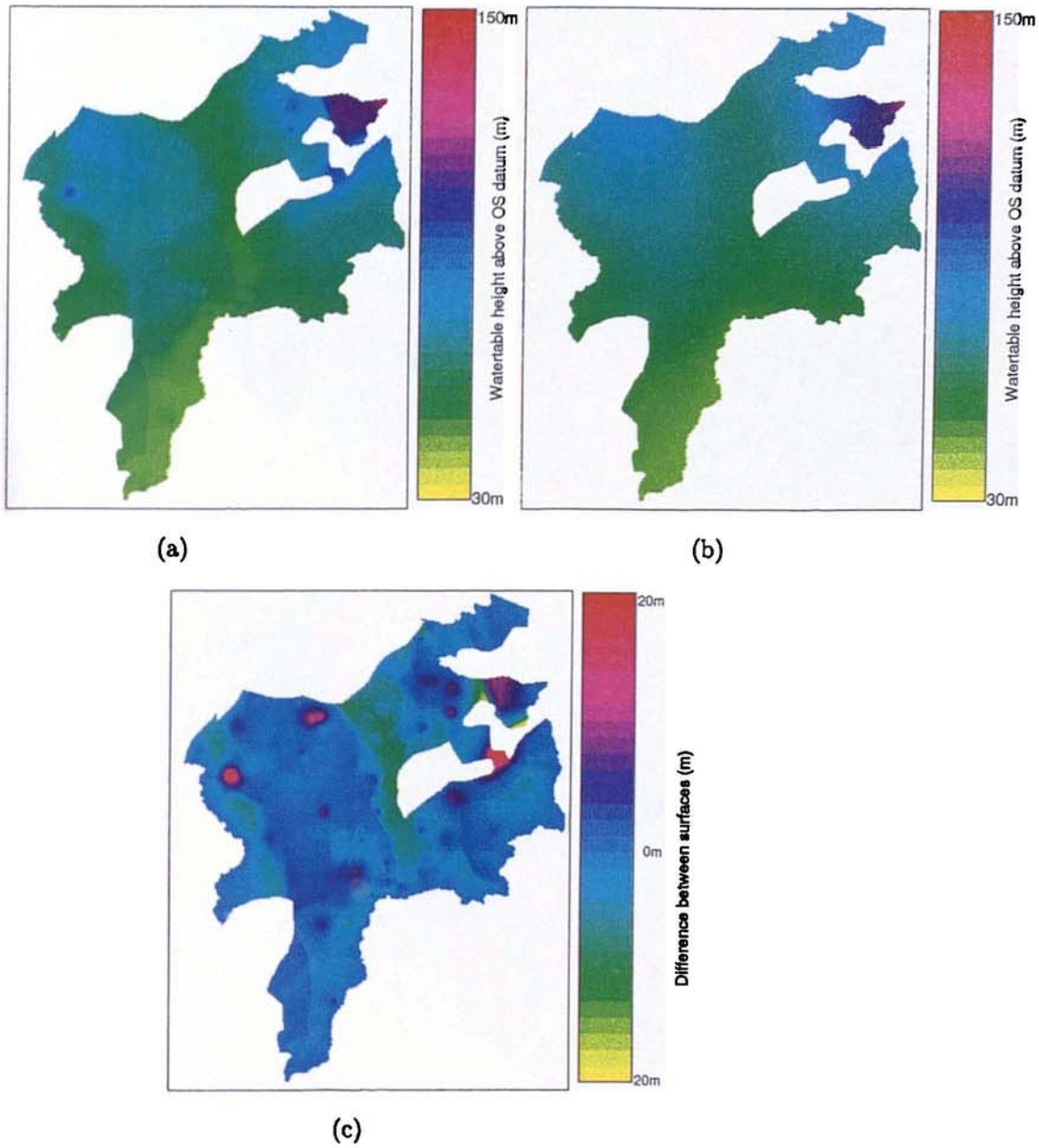


Figure 5.53: (a) The simple modelled surface, (b) the interpolated surface and (c) a raster of the difference between the model and the interpolation ($a - b$) generated with *r.mapcalc*.

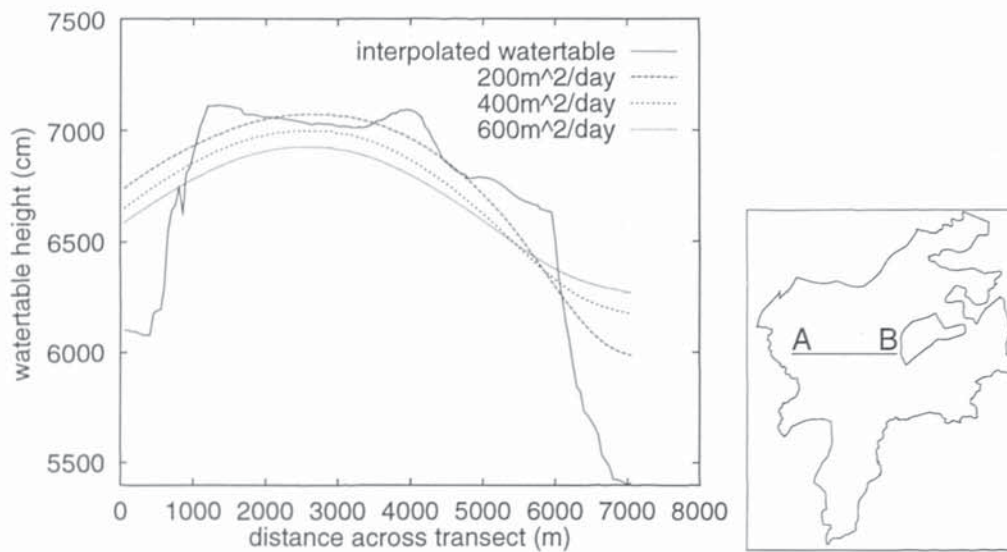


Figure 5.54: Cross-sections of modelled and interpolated watertable surfaces. Profiles extend from A to B on the location map.

Borehole	event	date	julian day
Lodgebank 2	Step Test	11.6.84	45862
Lodgebank 2	Constant Discharge test	18.6.84	45869
Lodgebank 1 & 2	Double Pumping Start	25.6.84	45876
Lodgebank 1 & 2	Double Pumping Finish	9.7.84	45890

Table 5.8: Timetable of Lodgebank pump tests.

the exercises into modelling the Lodgebank and Hodnet pump tests.

Lodgebank

The Lodgebank site has two abstraction boreholes. Lodgebank 1 was tested in March 1984⁹, in June and July 1990 Lodgebank 2 was tested and then both boreholes were pumped together. Table 5.8 gives the timetable of the second pump test event.

As touched upon in Section 5.5.9 The Lodgebank site is situated in a faulted region (Figure 5.28), Figure 5.29 shows an hydrogeologist's interpretation of the drawdown observations at the end of the double pumping test (NRA, 1985).

As with the previous modelling exercise the strategy has been to start with a simple model and add refinements as required.

⁹ a 12 hour step test and a 13 day constant rate test.

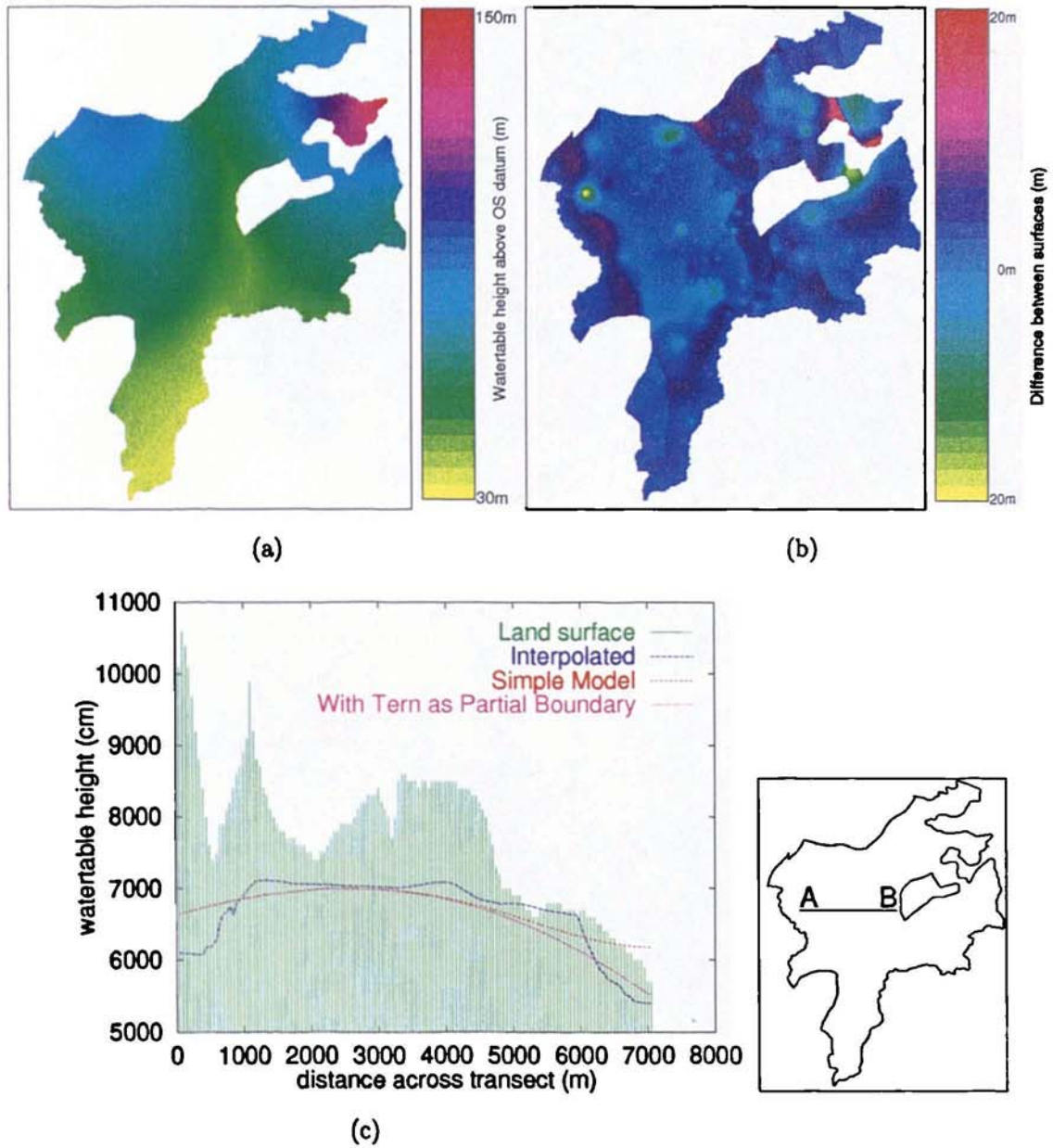


Figure 5.55: November 1990 watertable surface modelled with a partial boundary. (a) is the modelled watertable surface, (b) the difference between the modelled and interpolated surfaces and (c) cross-sections of interpolated and modelled surfaces. Profiles extend from A to B on the location map.

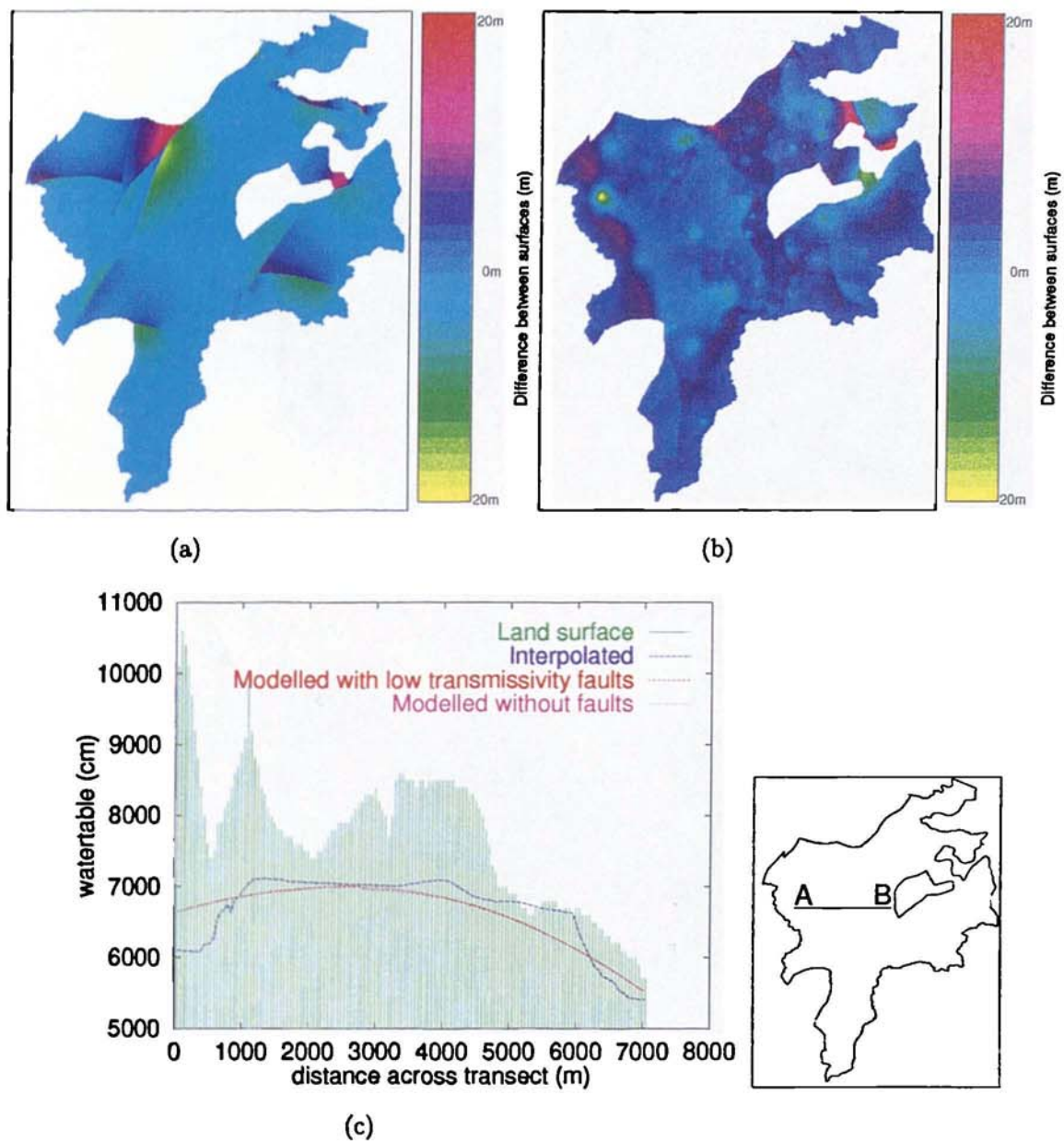


Figure 5.56: The groundwater model results after faults have been introduced.(a) the difference between the modelled surfaces with and without faults included, (b) the difference between the interpolated surface and the surface modelled with faults and (c) shows profiles of the modelled watertable under different assumptions. Profiles extend from A to B on the location map.

For this exercise it is the relative drawdown that is of interest rather than the absolute watertable height. Consequently as the gradient of the interpolated watertable is slight in this area it was decided to initially simplify the model by assuming the starting head is a horizontal plane. In this exercise no boundaries are specified, rather it is assumed that the only movement will be driven by the pumping so that if a sufficiently large area is chosen no edge effects will be apparent.

To model the effect of the faults the transmissivity of elements along the line of the faults are lowered. Start values for confined storage and transmissivity are taken from the pump test analysis. No value for unconfined storage could be calculated by the NRA so a value of 0.05 is taken on the basis of experience gained in the previous modelling exercise.

Figure 5.57 shows the output from the simple model overlaid with the NRA interpretation. Figure 5.58 shows the observed and modelled hydrographs at two sites within the region. The two sites chosen are the original Lodgebank pilot borehole, which is very close to the two abstraction boreholes and shows an immediate response and the monitoring well at Avenue Cottages (Figure 5.57). The well at Avenue Cottages is some distance 960m away from the abstraction boreholes, in the report (NRA, 1985) it is noted that during the Lodgebank No 1 pump test in March;

“ ... it is possible that an effect of Lodgebank's pumping was measured at Avenue Cottages although this was less than 0.1m. ”

During the second pump testing operation in June and July no effect was discernible at Avenue Cottages, the decline in Figure 5.58 is due to the overall groundwater recession in 1984.

The main differences between the observed and modelled levels is the different speed of the drawdown. The observed drawdown is much quicker and it is postulated that this may be the result of initially confined conditions occurring. This was tested by forcing confined conditions in the model (Figure 5.59) and the results indicate this is a possible explanation, although the modelled drawdown at Avenue Cottages is far larger than the observed. However, there is no evidence of geological features that would cause confinement in the reports or secondary material available. In relation to the hydrogeology of the Lodgebank region, the report of the 1984 test pumping (NRA, 1985) states.

“ ... Most of the area close to the Lodgebank site is free of drift or has a sandy drift cover, although an extensive area of sandy boulder clay occurs to the north of the site. This will produce a local perching horizon and account for the presence of a number of small pools in the area. This should, however, have no effect on the main aquifer as the sandstone watertable is well below the base of the drift. Unconfined conditions would therefore be expected throughout the area.

In addition to the drift, a series of minor, perched aquifers are associated with the sandstone bands within the Wildmoor sandstone. These are utilised for small scale farm supplies in the area e.g. Ivy Cottage and Stone and Hopton Cottages — again these should be independent of the main aquifer and, therefore, unaffected by the Lodgebank pumping. ”

Clearly some more complex system exists which is not captured by the model as it stands here.

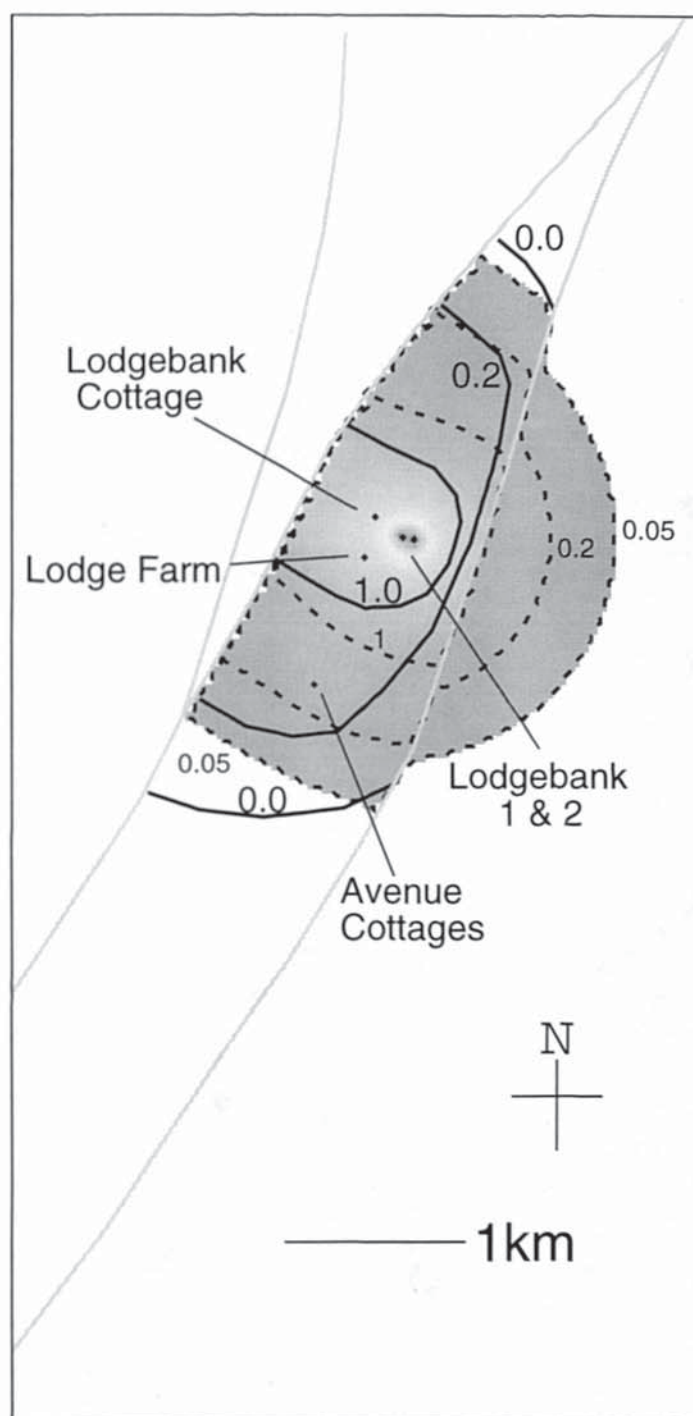


Figure 5.57: Results of modelling the Lodgebank boreholes. The solid black contours and “0.0” style numbers show drawdowns interpolated by hydrogeologists and the dashed black contours and “0.0” style numbers are generated use *r.contour* from the drawdown raster produced by the Modflow link.

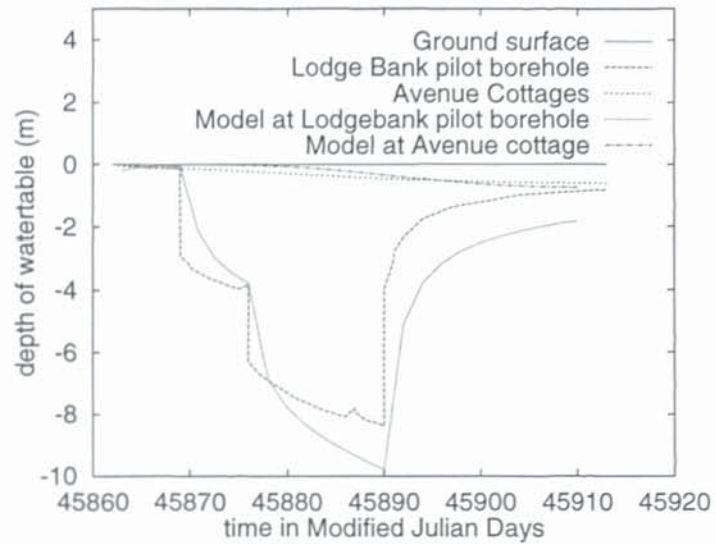


Figure 5.58: Observed and simple model simulation hydrographs of the Lodgebank double pump test. NB the pilot borehole was sunk during the pilot stage of development it is now used for observation.

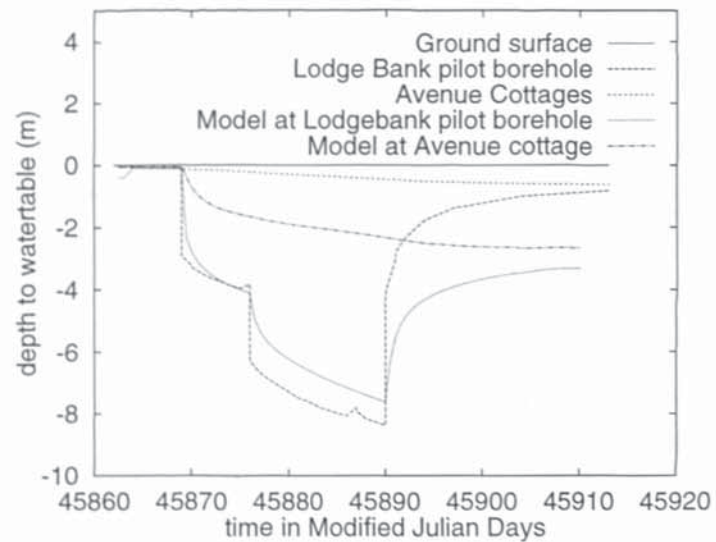


Figure 5.59: Observed and model simulation hydrographs of the Lodgebank double pump test. The model forces confined conditions at the beginning of pumping.



Figure 5.60: Hodnet abstraction borehole and surrounding observation sites.

Hodnet

During the Hodnet pump test in March 1984, 3 wells are reported to give a response to the pumping. These were Small Heath, Small Weobley deep tube well and the Hodnet observation borehole (Figure 5.60), at no other observation sites was a response noted.

As with all the modelling exercises the strategy is to start simple. Also, as with the Lodgebank simulation, it is the drawdown which is of interest, so the hydraulic head is treated as constant across the catchment. However, it is known that conditions can vary from confined to unconfined (see Section 5.5.8), so the distribution of drift must be accounted for. A cartographic modelling technique was used to allow a spatial non-varying hydraulic head to be used as the starting head. This meant that movement due to pumping would not be masked by movement due to the false hydraulic gradients in the initial starting head. Firstly, a watertable depth was interpolated for the start of pumping, this was normalised to zero by adjusting the aquifer top, i.e. aquifer top is equal to depth less the thickness of the clay drift (Figure 5.61). As an *r.mapcalc* query this is expressed as:

```
r.mapcalc aquifer_top="if(drift_type==310,depth-drift_thickness,depth)"
```

The river Tern is close, so the effects of leakage were also included in the model.

In this exercise an attempt was made to automate calibration. Various authors have written on this subject, and it is fraught with complications. The solution adopted here was to use an *ad hoc* script that performs a piece-wise search of the parameter space formed initially by the three variables transmissivity, and confined and unconfined storage coefficient — these were treated as constant across the area. The calibration script is shown in Figure 5.62. It is a

¹⁰ 3 is the code for clay in the drift type raster.

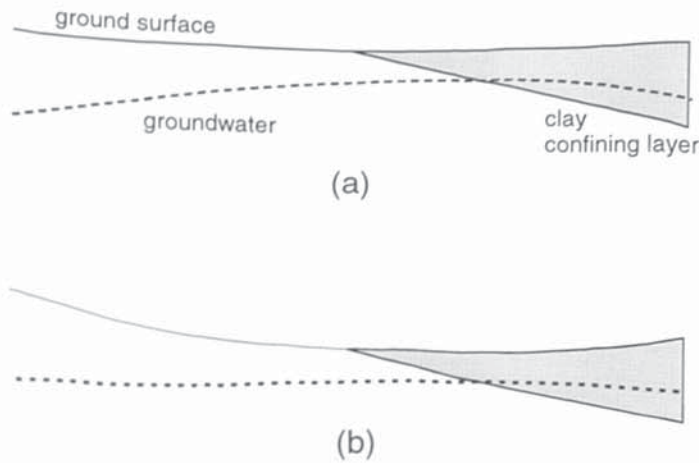


Figure 5.61: Cartographic modelling used to flatten the hydraulic head. (a) shows the unadjusted situation, (b) shows the horizontal hydraulic head and the adjustments necessary to other surfaces in order to maintain the relative depths

PERL script (Section 3.4.3), which firstly uses *r.mapcalc* to change the values of the rasters which describe the transmissivity and storage. Secondly, it calls Modflow. Thirdly another script *s.rms* (Appendix A.4.2) is used to calculate the root mean square (RMS) error between the modelled and observed drawdowns at several sites and several times.

A plot of transmissivity, confined storage and RMS is shown in Figure 5.63. For a wide range of values the model gives similar accuracy, there is an obvious region of higher accuracy and a region where the accuracy tails off. From this it can be seen that the model is fairly robust to changes in all these parameters over large ranges, and this is an encouraging result.

5.9 Evaluation

In terms of accurately and realistically representing groundwater movement, the large scale modelling of individual pump tests (Section 5.8.2) can be seen as more successful than the regional modelling (Section 5.8.1). In terms of exploring the issues and problems in linking a process model and GIS, in using the linked model and in gaining an understanding of aquifer characteristics, such as faults, river interaction and confinement all the exercises have been very useful.

This project has involved a significant amount of time in performing the modelling of the groundwater movement. There is no doubt that much more effort could have been expended in this area, indeed the construction of a comprehensive groundwater model of the Tern area

```
#!/usr/local/bin/perl

#open an output file and set last rms very high
open(OUTPUT,">hod.riv.dat");$lastmatot=999999999;

#for each transmissivity from 50m/s to 650m/s do:
for($strans=50;$strans<=700;$strans=$strans+100)
{

#for each unconfined storage from 0.04 to 0.20 do:
for($ustor=0.04;$ustor<=0.20;$ustor=$ustor+0.04)
{

#for each confined storage from 0.0005 to 0.0055 do:
for($stcr=0.0005;$stcr<=0.0055;$stcr=$stcr+0.001)
{

#generate rasters containing the current transmissivity and storages
'r.mapcalc hodnet_trans="if(isnull(bunter_fault.new),$strans,0)';
'r.mapcalc hodnet_ustor=$ustor';
'r.mapcalc hodnet_stcr=$stcr';

#write the modflow input files and run modflow
'mod.write -wr hodnet.river'; 'mod.run /tyne';

#extract the appropriate output
'mod.xtrct select=draw,1,3,1 select=draw,1,6,1 \
select=draw,1,12,1 select=draw,1,18,1 output=hodnet2 -j dir=/tyne/Tom';

#call the script r.rms to compare the model with the observations
$rm3='s.rms input=hodnet.3days map=hodnet2.draw.11.3';
$rm6='s.rms input=hodnet.3days map=hodnet2.draw.11.6';
$rm12='s.rms input=hodnet.3days map=hodnet2.draw.11.12';
$rm18='s.rms input=hodnet.3days map=hodnet2.draw.11.18';

#remove newline from rms and calculate total rms
chomp($rm3);chomp($rm6);chomp($rm12);chomp($rm18);
$rmatot=($rm3+$rm6+$rm12+$rm18)/4;

#print the results to a file
printf(OUTPUT"$strans\t$ustor\t$stcr\t$rm3\t$rm6\t$rm12\t$rm18\t$rmatot\n");

#remove the rasters
'mod.rm hodnet2.draw doit';
}
}
}

#close the output file
close(OUTPUT);
```

Figure 5.62: A script for performing a stepwise search of a parameter space to minimise the reported RMS error.

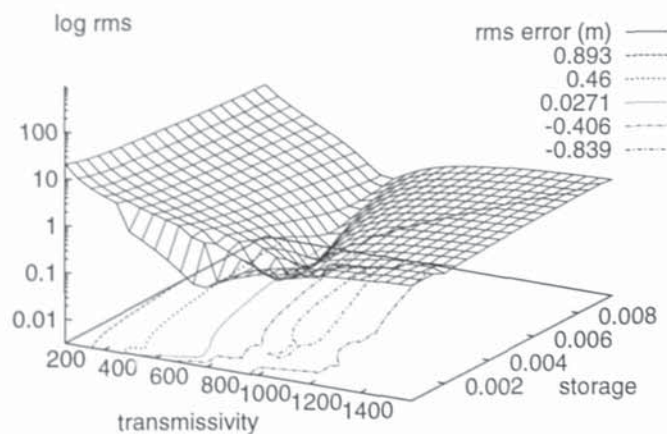


Figure 5.63: The parameter space formed by transmissivity and confined storage.

could have been the subject of a PhD research project on its own. The decision was made not to continue with the groundwater modelling further, but instead to use the lessons learnt to link in and use soil moisture and crop models. However, the experience of modelling along with personal communication with Professor Rushton (Rushton, 1996), have provided insights into improvements that might be made to the approach to modelling used here. It has also suggested other approaches to modelling that might be more amenable to achieving the aim of comparing pumping and non-pumping scenarios which ultimately provides a focus for this project.

This section is therefore intended to explore and summarise the lessons learnt from the process of linkage and using the model, and to evaluate the usefulness of the link. The success should be measured against the bench mark of Hedges's (1989) work.

5.9.1 The process of linkage

Achieving the link was relatively straight forward because of the basic compatibility of the underlying Modflow structures and GRASS rasters. Although a low level language C was used the GRASS libraries augmented its functionality so that programming was not completely from scratch. The issues raised by the linkage in terms of reconciling data models are discussed further in Chapter 8 in the light of the work undertaken with models of soil moisture and crops described in the next two chapters.

5.9.2 Using the GIS linked groundwater model

A major obstacle in the approach to regional modelling, followed here, came from attempting to interpolate a surface of groundwater heads and then using this as a starting point for modelling. Inevitably the major movement in the modelled surface is the movement to correct gross errors in these surfaces, i.e. where the watertable is specified too high the flow is away, where it is specified too low the movement is towards. The subtle movements caused by the driving forces of interest the annual recession, pumping etc. — are masked by these large movements. It is impossible, given the complexity of the system, to adequately specify a starting groundwater head surface and the model boundaries separately in such a way that there are major movement is caused by the real time varying conditions rather than the relaxation of the hydraulic head towards the steady state conditions of the specified model boundaries.

An alternative approach is to begin with a surface that is the steady state solution of the starting boundary conditions as determined during aquifer characterisation and expressed within the model specification. The surface can then be adjusted to more accurately represent the observed surface by adjusting the starting boundary conditions (e.g. by adjusting the hydraulic head specified in the connected rivers and the conductivity between the river and the aquifer), and not by directly manipulating the values in the surface itself.

Rushton was also concerned about the length of boundaries that had been specified as fixed-

head (Figure 5.12). In such a situation it is possible to achieve a model that appears to represent the surface well, but on examination of the mass balance the model surface is being maintained by implausible movement between the river and the aquifer. Nevertheless he did concede that river aquifer interaction in the Tern region is significant and should be included in the model.

The large scale modelling of the test pumping suggests another way in which the groundwater modelling might be approached. The simplifying assumption (i.e. that the starting head is flat) means that in effect these models represent the difference between the pumping and the non-pumping scenarios. In order to get a representation of the absolute position of the watertable it is necessary to add this difference to a surface representing the non-pumping scenario. Cartographic modelling provides a method of achieving this. Thus a process of generating a time sequence of surfaces can be envisaged. Firstly the meteorological record is searched for a representative year, for example a particularly dry year. Secondly data for that year is extracted from the groundwater readings and a sequence of surfaces generated using a method such as that in Section 5.5.2 or using a geostatistically based technique (Section 5.7). Thirdly a sequence of drawdown rasters is generated using the groundwater model, but making the simplifying assumptions of the test pumping exercises. Finally the drawdown surfaces are subtracted from the interpolated surfaces. the resultant surfaces could be used in a cartographic technique such as `possessivecitehedges:phd` or used as an input to a soil moisture process model, such as that described in the next chapter.

5.9.3 The Modflow–GRASS link

The conclusions that can be drawn from using the link are that:

- the link is a useful tool to predict the drawdown pattern for any pumping scenario that can be envisaged;
- that the link to GIS allows the drawdown patterns to be visualised with other data (e.g. Figure 5.64), and that this is a very useful facility when constructing and evaluating a model;
- the large number of GIS functions for display, capture, analysis and query were useful not only for assembling model input but also for evaluating, analysing and using the model output;
- the flexibility of the GIS allows other approaches to modelling drawdown to be easily implemented, such as scheme for modelling the difference outlined above (Section 5.9.2).

Images, such as Figure 5.64, illustrate how the spatial effects of pumping can be communicated and visualised, when the model is linked to a GIS.

5.9.4 Conclusion

Much work still needs to be done to adequately model the whole aquifer system in the way original envisaged, in order to distinguish between natural and artificial groundwater fluctuations. Indeed



Figure 5.64: Results of the Hodnet pump test model, displayed with a local map overlaid on the top.

it may be that it is not possible to adequately simulate the watertable movement across the aquifer to sufficient accuracy for this purpose — no examples were found in the literature where watertable position was required to such a level of accuracy, both in space and time.

This chapter has described the linkage of a groundwater model to a GIS, and the use of the resulting system. It represents the first attempt at applying the strategy outlined in Section 3.5.1 and improving on Hedges's (1989) static cartographic modelling technique (Chapter 4). It is a first prototype and as such inevitably involves the largest steps up the several learning curves that must be surmounted in order to understand and use the model and to link it with the GIS. Consequently it is the largest chapter in this thesis, as even in the next (Chapter 6) the lessons learnt greatly facilitate the task of linking the soil model and reduce the need for approaching the problems from first principles. For example, the choice of tools with which to construct the soil moisture model link (Section 6.5.1) is based on the experience of constructing the Modflow link (Section 5.4.3). Another example is the geostatistical techniques explored in Section 5.7, these could equally be applied to soil properties. However, this would largely be a repetition and the decision was made to look at a different area (i.e. stochastic simulation Section 6.7.2), but one which also addresses the fundamental problem of parameterising models, although from a slightly different paradigm. As an exercise in exploring the issues of linking existing GIS and process models, and in using the resulting system, the work described in this chapter can be considered successful.

Chapter 6

Soil moisture modelling

This section describes the soil moisture component of the system being modelled. Much of the soil science theory upon which this section is based is well established and dates back over several decades. This theory is first briefly reviewed, concentrating mostly on soil moisture and soil moisture modelling. The current research into the application of GIS for soil science and soil moisture modelling is examined. The model selected for this project, SWMS_2D, is described and the link between it and GRASS is discussed. Finally some of the modelling performed is presented.

6.1 Soil and soil moisture theory

The study of soil is a science in its own right with a vast literature, whilst soil mapping is the application of a complex methodology and requires considerable experience on the part of the mapper. These subjects are only briefly reviewed here. Soil moisture and soil moisture modelling are central themes of this thesis and will be reviewed in more depth.

6.1.1 Soil and soil mapping

Hilton (1986) defined soil as:

“ Soils are natural bodies whose properties are due to the integrated effect of climate and living matter acting on parent material as conditioned over periods of time. This may not be the most elegant of definitions but it does precisely convey the complexity of soils which occupy such a central place in biospheric processes. ”

Jenny (1980) expressed the same idea as a function:

$$\text{soil} = f\{\text{regional climate, organisms, relief, parent material, time, ...}\} \quad (6.1)$$

Jones (1975) gave a more basic definition:

“ The soil is any unconsolidated material directly below ground surface... ”

It is clear that the term soil encompasses a very complex idea, which has roots in geology, biology, climatology and hydrology, as well as agriculture, geography and civil engineering. The science of soil development and classification is termed pedology. The basic unit of soil science is the soil profile, and although all soils are different Figure 6.1 shows the profile of a typical soil and gives some commonly used terms.

When a soil is examined for classification, the first task is the description of the soil profile on the basis of the presence of characteristic soil horizons. The British Soil Survey uses a standard system to classify horizons, see for example Jones (1975).

During classification the presence and width of different soil horizons is noted. The properties of the individual soil horizons are recorded and include; colour, texture, structure, consistency (strength at various moisture levels) and organic content. In the UK only the top 1.5m is considered in the classification.

Texture is of prime importance and has significance for this project. Texture is a measure of the proportions of sand, silt and clay fractions. There is an international scale which defines texture the class sizes (see Figure 6.15), although the US Department of Agriculture (USDA) use slightly different divisions.

Horizons are divided into textural classes on the basis of the percentages of sand, silt and clay (Figure 6.2), and it is common for soils to be referred to merely by the class names, particularly when the soils have been classified in the field.

When mapping soils the basic unit is the soil series which was described by Robinson (1943) as:

“ group of soils similar in character and arrangement of the horizons of the profile and developed under similar conditions from one type of parent material. ”

However with regard to delineating soil series on a map Jones (1975) stated:

“ Map units and soil units are no longer regarded as synonymous ... A soil series is a defined single profile class but due to the inherent variability of soils it is not practicable to map the distribution of series. The map unit contains mainly soils of one series but has inclusions of one or more closely related profile classes which account for only a small proportion of its area. ”

For this project several sources of soil data were available at a variety of different scales and for the Tern region these are summarised in Figure 6.3.

At the smallest scale the whole area is covered by a regional soil map (Ragg et al., 1984) for which the basic mapping unit is the soil association. There are two types of soil association, in the first a single soil series is dominant (>50%) and there are a few ancillary series that will also be found in the association. The second type is more complex. There is no dominant soil series and there are generally more component soil series.

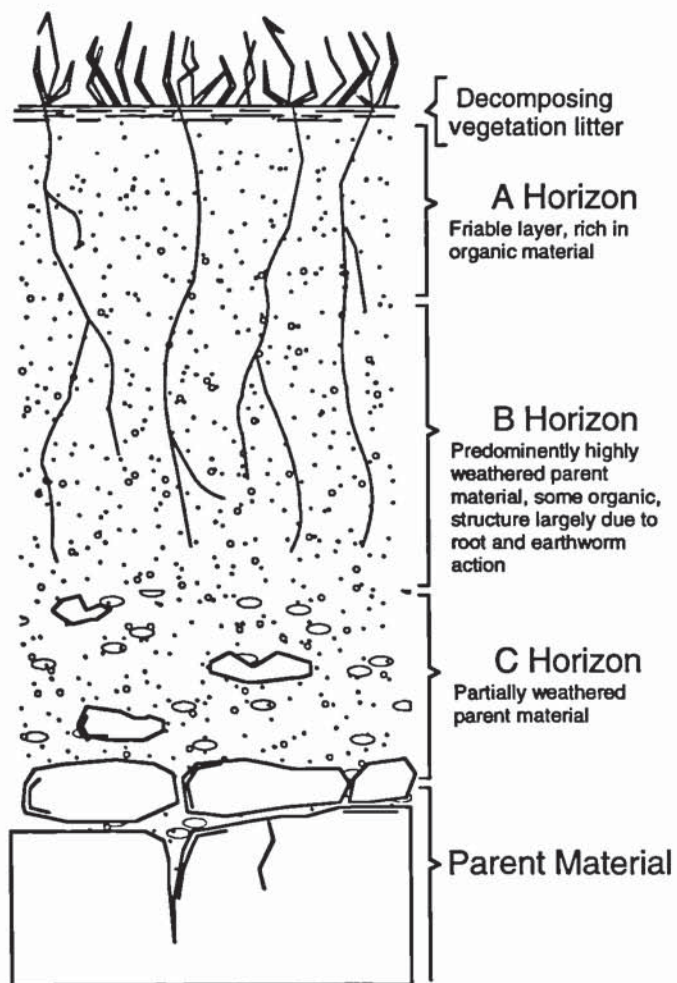


Figure 6.1: A typical soil profile.

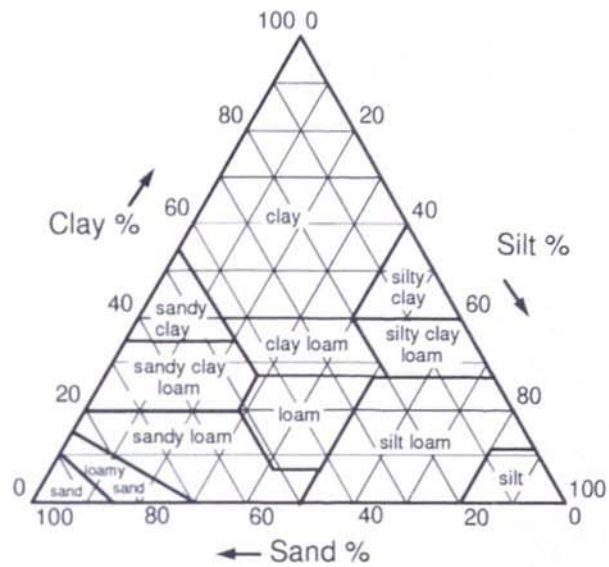


Figure 6.2: Classification of soils on the basis of particle size distribution.

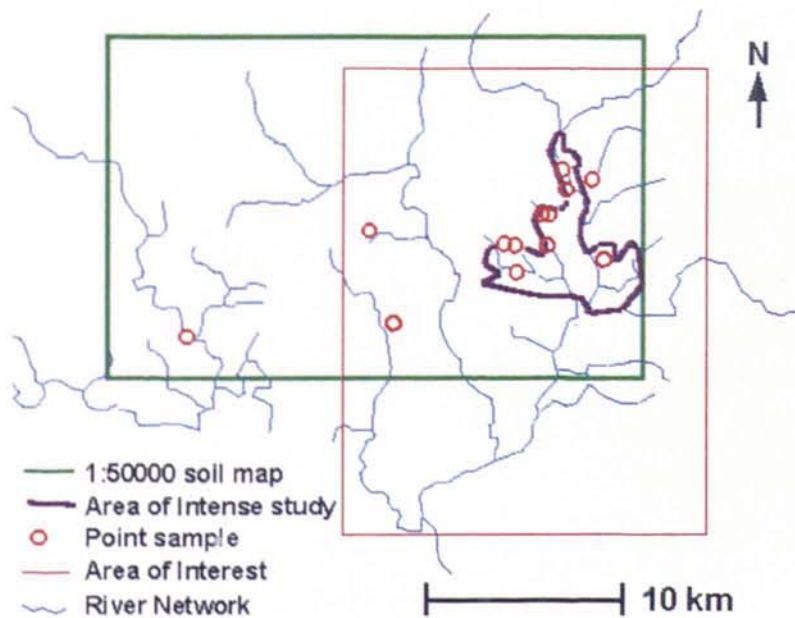


Figure 6.3: Availability of soil data in the Tern area.

At a scale of 1:63360 there is a soil map which covers a part of the region (Crompton and Osmond, 1954) and is based on soil series.

During the history of the Shropshire Groundwater Scheme several sites have been extensively studied, and there was also a very intensive survey over a small area in the Tern river valley (Soil Survey of England and Wales, 1982). At the scale of interest for this project all these data are effectively point samples. The point data takes a variety of forms: a detailed description of the soil profile obtained from a trial pit or augered sample; a description of soil hydraulic properties determined by direct measurement in the laboratory or field; or a record of the soil moisture regime over time. For the intensively studied area, the augered soil profile, obtained on an approximate 120m grid enabled the profiles to be classified into soil series, but with more detailed subdivisions being defined.

For a general discussion on soil see Pears (1985), in addition most soil survey memoirs include discussion on mapping methodology in the UK see for example Ragg et al. (1984), Crompton and Osmond (1954) or Jones (1975).

6.1.2 Soil moisture and soil moisture modelling

Moisture in soil exists in several distinct modes, as illustrated in Figure 6.4 and Figure 6.5. Moving from left to right in Figure 6.4 immediately adjacent to the soil particle surface is the immovable hygroscopic water. Beyond this, water exists in a film which can be divided into residual moisture (see below) and that water that can be lost to evapotranspiration, this latter water is held by matrix and capillary forces. Before water has been lost to evapotranspiration the soil is said to be at field capacity. Beyond the field capacity moisture is moved by gravity, water here is in a transient condition moving from infiltration at the surface to the saturated zone below. In the capillary zone the soil gets increasingly saturated with depth, but the water is held at tension i.e. less than atmospheric pressure. The tension decreases with depth until it reaches atmospheric pressure at the watertable below which the water is under increasingly higher pressure.

Several terms are commonly used in the literature and are defined below.

- *Soil moisture potential* (Φ), as with groundwater, soil moisture exists in a potential field. Feddes et al. (1978) defined the soil moisture potential at a point as Equation 6.2.

$$\Phi = \phi + \phi_g + (\phi_{osm} + \phi_{gas}) \quad (6.2)$$

where:

- Φ soil water potential at a point (L);
- ϕ matric potential some times called the *soil moisture pressure head*, caused by interaction between the moisture and the soil particles (L);
- ϕ_g the potential a point has due to its vertical displacement with reference to a datum point (L);

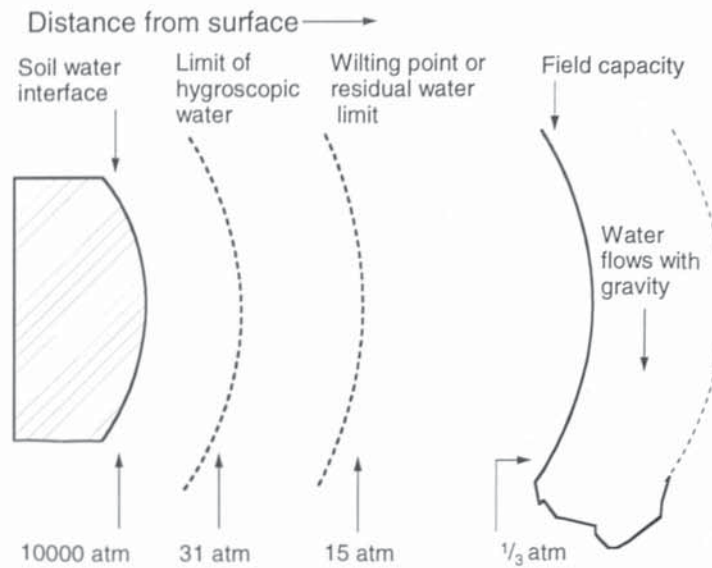


Figure 6.4: Modes of soil water after Pears (1985)

ϕ_{osm} — potential arising from osmotic forces (L);

ϕ_{gas} — potential arising from variation in gas pressure (L).

Generally ϕ_{osm} and ϕ_{gas} are considered to be insignificant.

- **Tension** is a term of convenience such that:

$$t = -\phi \quad (6.3)$$

where:

t — moisture tension at a point in the soil (L);

ϕ — matric potential (L).

Confusion can often arise between tension (h) or matric potential (ϕ) and total potential (Φ) or hydraulic head (h) (which include a gravity component). Matric potential is also often referred to as *pressure head* which is also frequently given the symbol h .

- **field capacity**— is the moisture fraction by volume of the soil that has been drained under gravity until drainage practically ceases, and no water has been lost to evapotranspiration. This is clearly a concept that is more of use as a general descriptor of conditions rather than a precisely measurable quantity. It might be taken to describe those conditions in a soil one or two days after a significant storm or irrigation but before a significant amount of water



Figure 6.5: Interaction of soil particles and water. After Wiesner (1970).

Table 6.1: Typical soil moisture properties as given by Marshall and Holmes (1979).

Head ϕ (cm)	potential (bar)
-1	-0.001
-10	-0.01
-100	-0.1
-1000	-1.0
-10,000	-10.0

Table 6.2: Soil moisture head and tension.

has been removed by evapotranspiration. Compare field capacity to Hedges and Walley's (1983) *field capacity profile* discussed in Section 2.2.1. Field capacity is often taken $\frac{1}{3}$ bar.

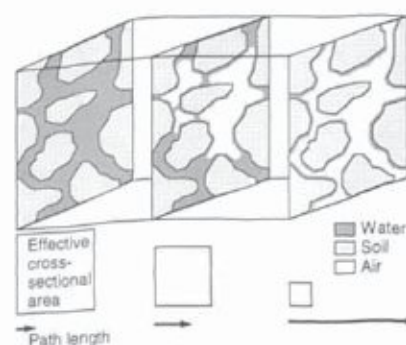
- *permanent wilting point* or *residual moisture* is that level of dryness beyond which a plant can no longer generate enough suction to remove any more moisture from the soil. Permanent wilting point is often taken as 15 bar.

Marshall and Holmes (1979) gave the properties for three different soils listed in Table 6.1.

- *soil moisture deficit* (SMD) is the difference between field capacity and the actual soil moisture status, it is commonly given as an equivalent depth of water.

In the literature it is common to see the value of soil moisture at a point or region in the soil expressed as soil water potential/tension or suction (ϕ or t) as an alternative to soil moisture content percentage. This is because soil moisture is often measured *in situ* using tensiometers (see Shaw (1993) for a description), and because a plant's ability to remove water, which relates to the suction that must be applied, is of importance in agriculture. Tension, matric potential and soil moisture content are often given as an *equivalent height* or *head* of water, e.g Table 6.2. There is not a linear relationship between water content and tension, and furthermore the retention curve is subject to hysteresis effects, Figure 6.7.a.

Childs (1969) illustrated the stages of drying in a soil with a picture similar to Figure 6.6. At stage 1 there is surface water and soil moisture is at above atmospheric pressure. The surface water drains away to stage 2, where the watertable is at the ground surface, tension has to be applied to remove further water to achieve stage 3. Further suction overcomes surface tension in the largest pores — this is the air entry point, stage 4. As more tension is applied increasingly



(a)

(b)

Figure 6.6: (a) Stages in a drying soil, after Childs (1969) and (b) change in effective cross-sectional area and path length as soil dries.

smaller pores will be emptied, stage 5. This process can take place in any part of the soil profile when moisture is removed.

Unlike groundwater modelling, the hydraulic conductivity in a soil at a point cannot be taken as constant through time. Conductivity changes as a complex function of soil moisture content. At saturation water can flow through the whole of the pore space. As the soil becomes increasingly dry pores empty and are no longer available for conducting moisture. At the driest stages moisture can only flow across the surface of soil particles and, as Hedges and Walley (1983) concluded, flow effectively ceases, Figures 6.6.b and 6.7.b.

So far it has been implicitly assumed that soil properties in a soil are spatially constant, this is clearly not the case so the above should be regarded as representing ideal soils — real processes are very complex.

6.1.3 Simple soil moisture models

The simplest soil models are conceptual mass balance models similar to those discussed in Section 3.2 and illustrated in Figure 3.1. These kinds of simple soil models are often included in runoff generation models. In the simplest of these the assumption is made that whilst there is a soil moisture deficit no runoff is generated. The resultant soil moisture content or tension is not the output parameter of interest, rather, the modellers are looking for a flood hydrograph. Vieux, Farajalla and Gaur's (1993) *r.water.fea* model is an example of this.

The MORECS (Meteorological Office Rainfall and Evaporation Calculation System) model is an example of a conceptual model from which the soil moisture deficit is one of the required output parameters. Examples of MORECS output data are included in Section 5.8.1, Figure 5.52. MORECS is a computer based system for estimating water balance variables from meteorological

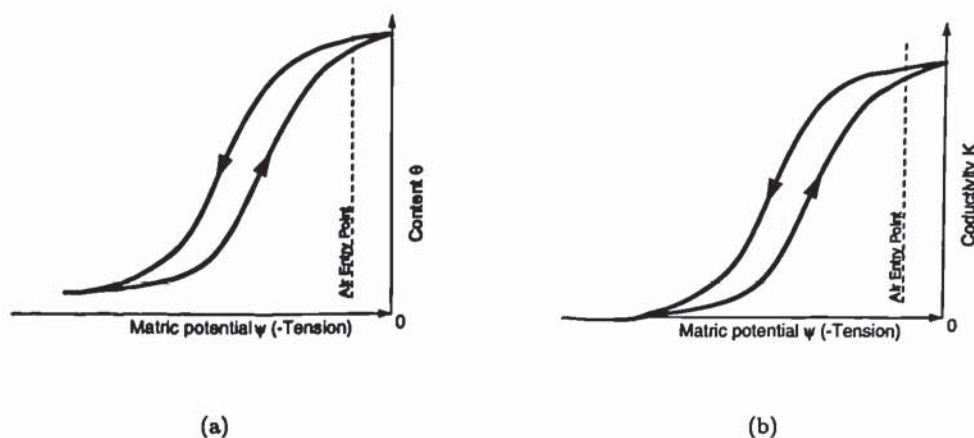


Figure 6.7: Typical (a) moisture-tension curve and (b) moisture-conductivity curves.

observations. Daily observation of the Penman variables (radiant energy, temperature, vapour pressure etc.) from approximately 150 non-uniformly distributed meteorological stations are interpolated to obtain daily values for 40km^2 squares covering England, Scotland and Wales. Potential Evapotranspiration (PE) is calculated for a range of land-covers using the Penman-Monteith equation (see Section 7.2.1). PE is converted to actual evapotranspiration (AE) by progressively reducing the rate of water loss from a maximum as the soil moisture reduces. Separate calculations are made for soils of high, medium and low water capacity. A calculation of the water balance of various cropped surface is made using estimates of the actual land-use for each 40 km^2 square. Finally the weekly averages of rain, sunshine hours, temperature, PE, AE, SMD and hydrologically active rainfall (rainfall available for run-off and aquifer recharge) are produced for each land use type in each 40km^2 . For further information see Meteorological Office (1981), Meteorological Office (1995) and Grindley (1967).

The model of van Keulen and van Laar's (1986) includes a soil moisture component which again is a conceptual model. There are two stores in the soil zone, the root zone and the soil beneath the root zone. The model allows the root zone region to advance into the unrooted zone as time progresses. Capillary movement is allowed for by using standard tables that depend on soil type and distance of the bottom of the root zone above the watertable to give the flux into the root zone. This model is described further in Section 7.2.5.

These kinds of models are of little use for this project, where the soil moisture profile and its response to watertable movement are of immediate interest. Models such as Walley and Hussein's (1982) approach the project requirements. Their model is still a conceptual model but the soil profile is divided into 4 layers, three of which act as classic conceptual stores and the fourth, the lowest, acts either as a sink for simulating deep percolation or allows the simulation of a shallow groundwater table.

6.1.4 Soil moisture flow models

Finite element techniques have been applied to soil moisture models. The flow region is divided into elements and flow is modelled using a finite element formulation of the Richard's equation. Equation 6.4 is Richard's equation in the vertical dimension:

$$\frac{\delta h}{\delta t} = \frac{1}{C_h} \frac{\delta}{\delta z} \left[K_h \left(\frac{\delta h}{\delta z} - 1 \right) \right] - S_h \quad (6.4)$$

where:

- h pressure head (L);
- t time;
- C_h differential moisture capacity $\frac{d\theta}{dh}$ (L^{-1});
- z vertical displacement (L);
- K_h hydraulic conductivity at pressure head h (LT^{-1});
- S_h sink term to account for root loss (LT^{-1}).

See Feddes et al. (1978) for a derivation of this equation.

From Equation 6.4 it is clear the major complication is the interdependencies, described in Section 6.1.2, Figure 6.7, between hydraulic conductivity (K) and soil moisture content (θ) with pressure head(h).

Unsaturated soil properties

Several authors have attempted to produce equations that describe unsaturated soil properties. van Genuchten (1980) used a model of the distribution of pore sizes to obtain a predictive equation for unsaturated moisture content and hydraulic conductivity, see Equations 6.5 and 6.6 (Figure 6.8).

$$\begin{aligned} \theta_h &= \theta_r + \frac{\theta_s - \theta_r}{(1 + |\alpha h|^n)^m} & \text{for } h < 0 \\ \theta_h &= \theta_s & \text{for } h \geq 0 \end{aligned} \quad (6.5)$$

$$\begin{aligned} K_h &= K_s K_{rh} & \text{for } h < 0 \\ K_h &= K_s & \text{for } h \geq 0 \end{aligned} \quad (6.6)$$

where:

$$\begin{aligned} K_r(h) &= S_e^{1/2} [1 - (1 - S_e^{1/m})^n]^2 \\ m &= 1 - \frac{1}{n} \\ S_e &= \frac{\theta - \theta_r}{\theta_s - \theta_r} \end{aligned} \quad (6.7)$$

where:

- K_h the hydraulic conductivity at pressure head h (LT^{-1});
- K_s the hydraulic conductivity under saturated conditions (LT^{-1});
- K_{rh} relative hydraulic conductivity at pressure head h (dimensionless);
- θ_h moisture content at pressure head h (dimensionless);
- θ_r residual moisture content (dimensionless);
- θ_s saturated moisture contents (dimensionless);
- α coefficient (L^{-1}),
- n exponent (dimensionless),
- S_e degree of saturation (dimensionless).

Brooks and Corey (1964) and Brooks and Corey (1966) developed a similar relationship for moisture content Equation 6.8

$$\frac{S_e}{S_e - 1} = \left(\frac{h_b}{h}\right)^q \quad \text{for } h > h_b \quad (6.8)$$

where

- S_e relative soil moisture content (dimensionless),
- h_b tension at air entry point L ,
- h pressure head L ,
- q pore size distribution index (dimensionless)

Evans (1992) notes that at high tensions van Genuchten's (1980) relationship reduces to an approximate form of the Brooks and Corey (1966) relationship. If van Genuchten's (1980) equation Equation 6.5 is rewritten for relative moisture content it becomes Equation 6.9.

$$S_e = \frac{1}{1 + \alpha h^{n+m}} \quad (6.9)$$

At large values of h

$$1 + \alpha h^{n+m} \approx \alpha h^{n+m} \quad (6.10)$$

it follows that

$$S_e = \left(\frac{1}{\alpha h}\right)^{nm} \quad (6.11)$$

By comparing Equation 6.11 and Equation 6.8 it is clear that the coefficient α can be considered as the inverse of the air entry point pressure head

$$\alpha = \frac{1}{h_b} \quad (6.12)$$

and

$$nm = q \quad (6.13)$$

Other authors have discussed the derivation of unsaturated soil properties and suggested similar relationships including Ritjema (1965) and Feddes et al. (1978). A common feature amongst the relationships is that hysteresis is usually ignored and, whichever author is consulted, there appears to be a core set of fundamental soil parameters that are required, these are:

- saturated hydraulic conductivity (K_s);
- saturated moisture content (θ_s);
- residual moisture content (θ_r);
- air-entry matric potential/tension.

These parameters are themselves not easy to estimate, and so several authors have investigated the possibility of estimating these from more easily measurable parameters such as bulk density and fraction of sand, silt and clay. Hedges (1989) derived some regression relationships for this purpose (see Section 2.2.3, Equations 2.5 and 2.4) using the sandy soils from the Tern region. Evans (1992) used three sets of secondary data from the US to derive statistical relationships for the important factors in van Genuchten's (1980) model.

For moisture content at 15 bar, which many authors consider to be a reasonable approximation of residual moisture or the wilting point, Evans generated;

$$\theta_r = 0.0042C + -0.000000016C^2 - 0.00036Sd + 0.044 \quad (6.14)$$

for moisture at $\frac{1}{3}$ bar (an estimate of field capacity);

$$\theta_{fc} = 0.005C - 0.000023Sd^2 - 0.248 \quad (6.15)$$

and for van Genuchten's (1980) α and n Evans generated;

$$\alpha = 0.0018Sd_{vc} - 0.000004Sd_c + 0.0025S^2 + 0.026 \quad (6.16)$$

$$n = 0.066Sd_f - 0.0026Sd_f^2 + 0.000032Sd_f^3 - 0.00043Sd^2 + 0.0000049Sd^3 + 1.33 \quad (6.17)$$

where:

C percentage clay;

Sd — percentage sand;
 Sd_{vc} — percentage of very coarse sand;
 Sd_c — percentage coarse sand;
 Sd_f — percentage fine sand;

Evans erroneously considers van Genuchten's (1980) α to be dimensionless, whereas inspection of Equation 6.5 shows the dimensions of α are L^{-1} , consequently it is hard to know how to interpret the results of Equation 6.16, but as the majority of Evans's (1992) work was performed at the cm scale, cm^{-1} was deduced to be the units for α .

Evans (1992) tested his relationships against Hedges (1989) experimental columns (described in Section 2.2.3) and achieved reasonably satisfactory matches.

The root sink zone

The subject of plant and crop modelling is dealt with in Section 7. However, most soil models deal with plants by treating them as a boundary condition.

In Equation 6.4 the effect of evapotranspiration is included as a sink term. In the model SWATR Feddes et al. (1978) used a simple model to describe how evapotranspiration varied with head (Figure 6.9).

6.2 GIS and soil

As in other disciplines GIS have begun to find their way into soil related research projects, often for mapping, but also for other aspects of the science. Burrough's (1986) book, arguably the classic text for GIS, was written from a soil resources background and many examples in the book reflect this.

Gessler, McKenzie and Hutchinson (1996) summarised work performed on the spatial prediction of soil properties using GIS. The basic assumption in the work they describe was that:

“ Spatial analysis of a digital terrain attribute, the compound topographic index (cti), was used to define the local catenary/landscape population and catenary environmental gradients for allocating field sample locations. The assumption was that the stratifying variable, cti, captured the range and relative distribution of variation in the landscape for the attributes of interest. ”

Fundamentally they are using several different statistical methods to derive relationships between variables of interest and those variables that can be spatially well defined, and thus produce cartographic models which describe the spatial variation of the variables of interest. The principal independent variables are topographic defined from the digital elevation models, and they also include geological and climatic information.

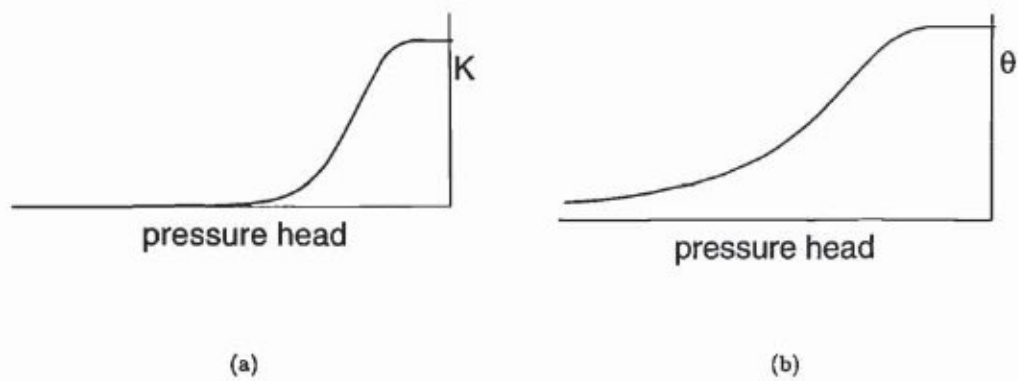


Figure 6.8: van Genuchten's (1980) soil properties model, relationship of pressure head (-tension) to (a) conductivity and (b) moisture.



Figure 6.9: Feddes et al.'s (1978) model of the root sink term. h_1-h_2 root extraction increases linearly as conditions become less saturated. h_2-h_3 root extraction is at a maximum. h_3-h_4 root extraction decreases linearly with increasing low moisture and higher tensions. h_4 is wilting point.

GIS are not ideal for handling soil data. Ramlal and Beard (1996) recognised a fundamental weakness that neither the vector model nor the raster model is adequate to represent the complex spatial variation of soil. They suggest a *mixed variation model* for tackling this problem. This is an attempt to generate a combined raster-vector object.

Rogowski and Engman (1996) used a GIS framework to correct a temporal sequence of synthetic aperture radar (SAR) using a few point measurements of soil within a geostatistical simulation method to remove the effects of vegetation, surface roughness and aspect to give an image of the soil moisture status of the soil.

The one area in which GIS has been significant is in the study of soil erosion. In Section 3.3.3 a very simple cartographic model was presented to predict the spatial distribution of vulnerability to erosion and many of the studies use this principle, although in a much more sophisticated way.

Meijerink et al. (1996) compared three approaches for modelling erosion via estimation of flow accumulation within a GIS. All three approaches use flow accumulation derived from a DEM.

Ludwig et al. (1996) used GIS to predict the spatial distribution of concentrated flow erosion in an agricultural catchment. Their work was performed at a very large scale, and used such features as tillage direction and *agricultural linear features* (i.e. field boundaries etc) to predict in-field flow fields and the location of flow concentrating structures.

Ellis (1996) used a GIS framework to handle the data from multiple sources so that artificial techniques (neural networks and decision tree analysis) could be applied to the data in order to improve the prediction of the spatial distribution of erosion at a regional scale (30m grid).

The trend of linking GIS and environmental process models can be seen in soil studies with erosion studies leading the way. Savabi et al. (1996) used the GRASS GIS to parameterise the physical erosion model WEPP (Water Erosion Prediction Project, USDA, 1995). Bian et al. (1996) linked the GIS Arc-Info to the model SWAT (Soil and Water Assessment Tool), which is designed to simulate water and soil yields from large rural catchments.

One of the few examples of the linkage of a soil moisture model, that is not designed for erosion or run-off simulation, with a GIS is Vermulst et al. (1996) who linked a vertical 1-dimensional unsaturated flow model to a groundwater model via a GIS in a similar manner to that envisaged in this thesis. Their work is performed on a much smaller scale (over the whole of Holland) and the unsaturated model is applied to plots of approximately 500×500m across the country, the groundwater model has elements of 1 to 5 km². The model is used for assessing the effects of national water use policy on, for example, desiccation, eutrophication of surface waters etc.

The only other comparable example of linking a vertical 1-dimensional unsaturated flow model to a GIS found in the literature was the research of described by Vaughan et al. (1996) (incidentally from the same research institution that produced SWMS.2D). They linked *Unsatchem*, a 1 or 2-dimensional unsaturated water flow, multi-component chemical transport, CO₂ transport and heat transport model with the GIS Arc-Info. Their concern was modelling that part of the carbon dioxide cycle in the soil zone.

6.3 Accuracy, sensitivity and uncertainty of soil moisture models

Several authors have dealt with the perennial issues of accuracy, sensitivity and uncertainty, with regard to soil moisture modelling

Clemente et al. (1994) compared three similar models on the same field data. These models were SWATRE, an extended version of SWATR, (Belmans, Wesseling and Feddes, 1983), LEACHM, Leaching Estimation and Chemistry Model, (Wagenet and Hutson, 1989) and SWASIM, Soil Water Simulation Model, (Hayhoe and De Jong, 1982). These models are conceptually the same so the differences between the outputs are, therefore, due to differences in finite difference/element formulation, the methods of solving the simultaneous equation and the deployment of sub-models such as the root sink. Clemente et al. concluded there was little difference between the models none of which consistently outperformed the others.

Zepp and Belz (1992) reviewed the problems of soil moisture modelling and pointed out:

“ A major problem impeding the application of physically based simulation models is the determination of many input variables and functions; modelling soil moisture conditions has been confined mostly to selected sites. ”

This problem is well illustrated by the soil modelling exercise in Section 6.6.

6.4 SWMS_2D

Several models were examined for use in this project. The criteria for the selection were:

- capable of modelling soil moisture variation down the profile.
- able to model the effects of a varying watertable.
- easily linked to a GIS, and able to work well with the other models in the system.

The finite element approach was considered appropriate and two such models were readily available. The models SWATR (Feddes et al., 1978) and SWMS_2D (Simunek et al., 1994). These models are very similar, as demonstrated by a comparison reported in Simunek et al. (1994). The model SWMS_2D was chosen as the source code is in the public domain and available in a computer readable form. A version of the SWATR source code is published in Feddes et al. (1978) but a computer readable form was not obtainable.

6.4.1 Practical features of SWMS_2D

SWMS_2D was written by the US Salinity Laboratory, a part of the US Department of Agriculture. It was written in FORTRAN 77 and is very portable (executable versions were compiled on both

UNIX and DOS platforms). The model is capable of simulating unsaturated flow and solute transport in 1 or 2 dimensions, although for this project only flow modelling in one dimension is required.

As with Modflow, SWMS_2D uses text files to read input data from. There are three input files: the basic input file describes the basic model parameters (the soil properties, the number and length of iterations etc.); the grid file describes the finite element grid; and, the atmosphere file describes all the time varying boundary conditions (precipitation, groundwater level, root extraction etc.).

One characteristic of SWMS_2D that has become apparent through extensive use is that it appears to lack routines for checking basic errors in the input files. When reading input data SWMS_2D does not check further than the type checking performed by library functions when reading input (i.e. the library function READ will generate an error message and terminate the program when a text string is encountered when an integer is expected). Furthermore SWMS_2D does not generate any useful error messages. In its current state it can be considered the *bare bones* of a modelling system. This can make it very frustrating to use and tracking errors in the input files can involve considerable detective work and trial and error.

6.4.2 Characterisation of soil properties in SWMS_2D

SWMS_2D uses a modified form of van Genuchten's (1980) equation to generate the tension/content and tension/conductivity relationships (Section 6.1.4). In the modified form four extra parameters are used. An extra head parameter (h_k) and a conductivity parameter (K_k) are added to modify the tension/conductivity curve illustrated in Figure 6.8.a. The main feature of the modified model is that it allows for saturated conductivity to be maintained until air entry point (h_s) rather than at zero head. Similarly two extra content parameters are added (θ_a and θ_m). These modify the tension/content curve as illustrated in Figure 6.10.b and maintain saturated content until air entry point.

$$\begin{aligned}\theta_h &= \theta_a + \frac{\theta_m - \theta_a}{(1 + |\alpha h|)^m} \quad \text{for } h < h_s \\ \theta_h &= \theta_s \quad \text{for } h \geq h_s\end{aligned}\tag{6.18}$$

$$\begin{aligned}K_h &= K_s K_{rh} \quad \text{for } h \leq h_k \\ K_h &= \frac{(h - h_k)(K_s - K_k)}{h_s - h_k} \quad h_k < h < h_s \\ K_h &= K_s \quad \text{for } h \geq h_s\end{aligned}\tag{6.19}$$



(a)



(b)

Figure 6.10: Modified form of van Genuchten's (1980) soil properties model, relationship of pressure head (-tension) to (a) conductivity and (b) moisture.

$$\begin{aligned}
K_{rh} &= \frac{K_h}{K_s} \left(\frac{S_e}{S_{ek}} \right)^{1/2} \left[\frac{F(\theta_r) - F(\theta)}{F(\theta_r) - F(\theta_h)} \right]^2 \\
F(\theta) &= \left[1 - \left(\frac{\theta - \theta_a}{\theta_m - \theta_a} \right)^{\frac{1}{m}} \right]^m \\
S_e &= \frac{\theta - \theta_r}{\theta_s - \theta_r} \theta_r \\
S_{ek} &= \frac{\theta_k - \theta_r}{\theta_s - \theta_r} \theta_r \\
m &= 1 - \frac{1}{n}
\end{aligned} \tag{6.20}$$

where:

- K_h the hydraulic conductivity at pressure head h (LT^{-1});
- K_s the hydraulic conductivity under saturated conditions (LT^{-1});
- K_k hydraulic conductivity at h_k (LT^{-1});
- h_s pressure head at air entry point $h_s \leq 0$, (L);
- h_k pressure head which produces the conductivity K_k (L);
- K_{rh} relative hydraulic conductivity at pressure head h (dimensionless).
- θ_h moisture content as pressure head h (dimensionless);
- θ_a moisture content, such that $\theta_a \leq \theta_r$ (dimensionless);
- θ_m moisture content such that $\theta_m \geq \theta_s$ (dimensionless);
- θ_s saturated moisture contents (dimensionless);
- α coefficient (L^{-1}), as explained in Section 6.1.4 can be taken as the inverse of the air entry pressure head (h_s);
- n exponent (dimensionless);
- S_e degree of saturation (dimensionless);
- S_{ek} degree of saturation using θ_k (dimensionless).

SWMS_2D uses a modified form of Feddes et al.'s (1978) model of the root sink term (the original is illustrated in Figure 6.9). Instead of a single h_3 term to mark the head below which roots extract at the maximum root two terms are given h_{3u} and h_{3l} which mark the upper and lower limit to a sliding scale along which h_3 can vary linearly. At a high potential transpiration rate (T_u), h_3 is set to h_{3u} , at a low potential transpiration rate (T_l) h_3 is set to h_{3l} . Between these two transpiration rates h_3 varies linearly between h_{3u} and h_{3l} , Figure 6.11.

It is a weakness of SWMS_2D that the depth of the root zone is not allowed to vary during the course of the model. It does, however, allow evaporation to be treated separately from transpiration, (evaporation is assumed to come from the surface nodes as indicated by a node code in the grid specification, whilst transpiration is distributed by a user defined function through the root zone again included in the grid specification). The result of this is that it is possible to model the change from bare earth to covered in a crude way. Potential evaporation and transpiration rates are defined by the user for each period. These are modified during the model run according to the moisture status of the soil.



Figure 6.11: the modified form of Feddes et al.'s (1978) model of the root sink term used by SWMS_2D.

6.5 The link between SWMS_2D and GRASS

The strategy followed to link SWMS_2D to GRASS is outlined in Section 3.5. The linkage was performed at two different levels. A basic link between the model and GRASS was achieved by sampling the rasters in the GRASS database at a single point and running the model on this data. This link was then embedded within a higher level controlling process which calls the link repeatedly at different points over the area of interest.

6.5.1 Basic link

Figure 6.12 shows the basic link between SWMS_2D and GRASS.

SWMS_2D has a lot more capabilities than are required by this project. It can model in two dimensions and it can model solute transport but since they are not required, in order to save time these capabilities have been ignored and the link has been constructed merely to exploit the SWMS_2D vertical 1-dimensional flow modelling capabilities. A dataset is designated by the user which consists of pointers to rasters which describe the different properties of the soil as they vary across the landscape, and lists of rasters which describe boundary conditions as they vary through time.

The link was constructed in 'C' code, using the GRASS libraries. Unlike the Modflow link (see Section 5.4.3) the SWMS_2D to GRASS link consists of a single program *r.swms*. The GRASS's

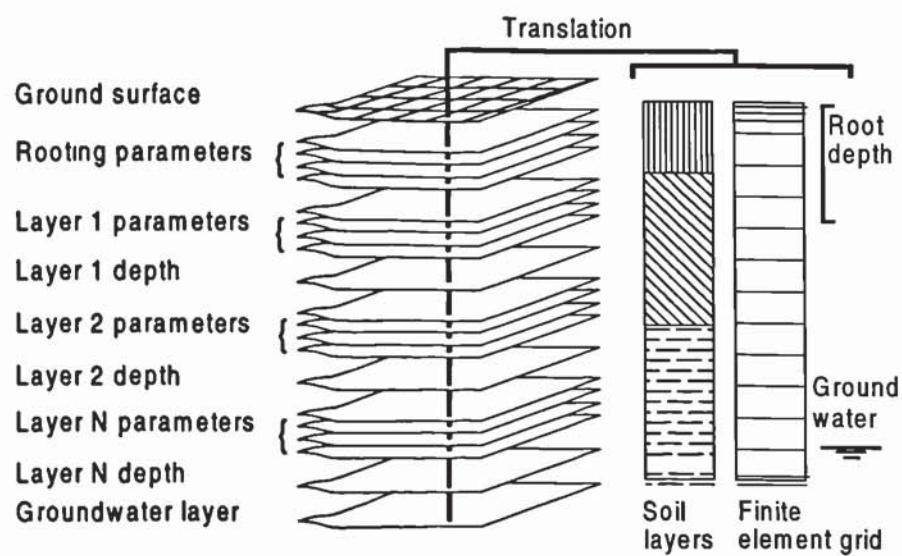


Figure 6.12: Basic link between SWMS_2D and GRASS.

visual ask (VASK) library, which was used to construct *mod.edit* (see Appendix A.1.1), was not used for *r.swms*, instead the GRASS parser library (Shapiro et al., 1993) was the principal tool for constructing the interface (see Appendix A.1.1).

The parser has two distinct modes; firstly if the command name is entered on its own then the program enters an interactive mode and asks for the data it needs. The alternative is to enter the options on the command line. Most of the GRASS core programs operate in this fashion, with the advantage that the programs can be embedded within a script.

r.swms follows the strategy depicted in Figure 3.9. A dataset is constructed which points at those GRASS database objects (rasters in the case of *r.swms*) which describe the model parameters and boundaries. From these *r.swms* can (if the *-w* flag is set) write these to the input files needed by SWMS_2D and can then call SWMS_2D to run (if the *-r* flag is set).

As an alternative to answering questions or including all options on the command line, the user can call up a simple interface (by setting the *-e* flag) with which to directly edit the dataset. This interface has been constructed using a simple text editor rather than VASK as *mod.edit* was (see Section 5.4.3). As with *mod.edit* the user can just specify a number instead of a raster, if the property being described does not vary spatially. Unlike *mod.edit* the user does not need to state that a number rather than a raster has been given, *r.swms* will automatically check for this.

6.5.2 The controlling process

There is a need to apply the model repeatedly across the landscape for all those areas in which the soil moisture might be affected by watertable movement. A PERL script *r.swms.region* (Appendix A.4.3) was constructed using the PERL-GRASS library (Section A.4). It was designed to apply the basic link above pixel by pixel across the landscape.

A significant practical problem is the time taken to complete all the model runs. Each model run is typically about 1-5 minutes. If this is multiplied by the number of pixels (typically several thousand) and it is seen that the task could run into days.

One of the options that can be chosen with *r.swms.region* is to use a raster as a mask to exclude all areas which are not needed. A suitable masking raster would be one constructed by application of Hedges's (1989) equation for delineating vulnerable areas (i.e. Section 4.2, Figure 4.3).

Another way of minimising the time taken for computation is to perform the model runs in parallel. Each model run is considered independently in areas where watertable movement might affect the soil moisture profile and lateral flow is likely to be insignificant. For this project several UNIX computers were available, and it is possible under the UNIX paradigm for several users to use the same computer — one person logged in and interacting directly with the computer and others logged in remotely. *r.swms.region* was written to spread the modelling effort over a number of computers using a 'machine-list' option. This required some quite sophisticated process management. The controlling process has to spawn child processes on the different machines, with each child process controlling the writing of the model input files and calling SWMS_2D to run

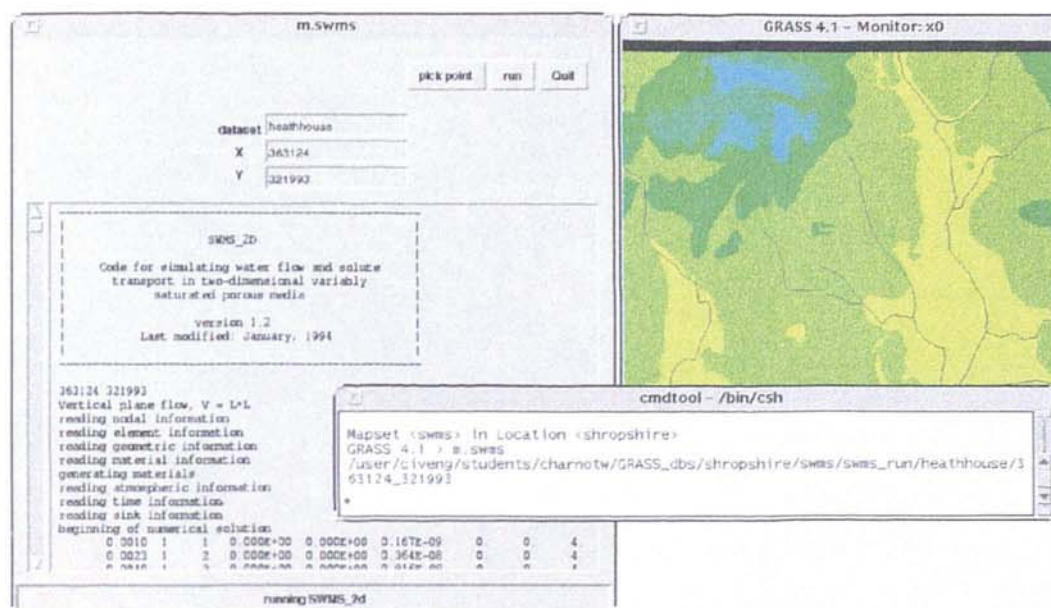


Figure 6.13: A Tcl interface to the SWMS_2D link.

on a particular pixel. When a child process ends the controlling process spawns a new process to model the soil moisture on the next pixel in a region. The code for *r.swms.region* is documented in Appendix A.4.3 and the user guide is in Appendix C.2.

6.5.3 A graphical controlling process

One advantage of the two level approach adopted is that the basic SWMS_2D link can be embedded within some other controlling process. As an exercise the *Tool Command Language* was used to construct a graphical interface, Figure 6.13. The idea behind this link was that the mouse was used to select a location by clicking on a GRASS monitor where the model was to be run.

6.6 Modelling with SWMS_2D

Figure 3.11 shows how it was envisaged that the soil moisture model would be applied over the entire region. However, it was clear that for an academic project and one that is intended to explore the issues and considerations of using GIS linked models, more valuable experience and knowledge could be obtained by seeing how the model performed at a single well observed site. The Heath House site was chosen since it was one of the sites originally investigated by Hedges (1989), Figure 6.14. During 1979, before the Local Public Inquiry, it was at the Heath House site that Hedges identified a drop in soil moisture due to watertable drawdown, Section 2.2.2.

The Heath House site is within the Newport soil series, and as part of his investigation Hedges



(a)



(b)

Figure 6.14: The location of the Heath House site.



Figure 6.15: Standards of particle size classes , ISSS: International Society of Soil Science, USDA: US Department of Agriculture and BSI: British Standards Institution. After Shaw (1993).

monitored soil moisture at five neutron probe access tubes at increasing distances from the abstraction borehole together with the watertable level at three wells. The monitoring program lasted most of 1979 and into 1980. On completion of the monitoring program a pit was dug at each tubewell and samples taken and thus there is very good characterisation of the top of the soil profile at these points. The full results of this investigation was given in Hedges (1989).

As Hedges documented the particle size distributions of the profiles at each observation tube at the Heath House site. or this modelling exercise the equations Hedges's (1989) produced were preferentially chosen to estimate the hydraulic properties of the soil. Where these fail then the equations generated by Evans (1992) were used. The problem with using Evans's (1992) equations is that they have been generated using particle size classes as defined by the US Department of Agriculture (USDA). Hedges analysed his samples according to standards laid down by the British Standards Institute (BSI) Figure 6.15. Where Evans's (1992) equations are used Hedges's (1989) particle size distributions have been resampled to the USDA particle size classes. This was done assuming a uniform particle size distribution across a particle size class.

Hedges documented the soil properties of the access tubes at Heath House down to a depth of approximately 2m. At tubewell 5 the watertable was dropped from a depth of approximately 3m to one of 6m. The soil properties of the bottom part of the profile can only be estimated but, as

layer	depth (cm)	Gravel	Co Sand	Med Sand	Fine Sand	Total Sand	Silt	Clay
1	0 – 30	4.91	1.35	31.25	38.96	71.56	19.53	4
2	30 – 51	3.24	1.72	34.41	40.17	76.3	17.86	2.6
3	51 – 95	7.3	2.59	34.52	42.98	80.90	12.41	0.2
4	95 – 117	35.02	2.01	31.12	31.12	27.35	60.48	4.5
5	117 – 185	0	0.02	32.41	58.18	90.61	7.49	1.9

Table 6.3: Heath House, tube 5 particle size distributions.

it was apparent to Hedges that the parent rock had been reached, the properties from the lowest sample level are assumed to persist to the bottom of the profile.

The access wells were located in a field of sugar beet and laid out as illustrated in Figure 6.16. During his investigation Hedges noted roots to a depth of 1.80m and on examining the soil moisture records stated:

“ The soil moisture profiles ... exhibit moisture loss to a depth of 1.8m which can only be attributed to root abstraction. ... For the soil conditions at the Heath House site, a rooting depth for sugar beet of 1.8m can be safely accepted. ”

Because of a weakness in SWMS_2D a constant root depth of 1.80m had to be assumed throughout the season, an unsatisfactory arrangement at best. Potential evapotranspiration of a typical year was extract from the MORECS data and this was partitioned between evaporation and transpiration by simply adjusting the root abstraction model to favour a maximum abstraction from the surface of the soil decreasing linearly down to 1.80m.

The data from Hedges (repeated in Table 6.3) was encoded into the SWMS_2D input files along with other data and the assumptions discussed above. Figure 6.17.a shows the progress of the soil moisture through the season.

Figure 6.17.c shows the predicted development using the Hedges's (1989) and Evans's (1992) regression equations to generate the input values.

As a comparison another dataset was generated by taking example values from the literature (principally from the SWMS_2D manual, Simunek et al., 1994) and manipulating these basically in a trial and error approach — the results of these are shown in Figure 6.17.b. Figure 6.18 shows a more detail view of the observed and simulated development of the soil moisture profile. If Figure 6.18 is viewed from the side then each graph shows the soil moisture down the soil column. The top row shows the observed soil moisture regime through the test pumping. The second and third rows show the modelled soil moisture regime using regression equations to parameterise the model. The forth and fifth rows show the modelled soil moisture regime using trial and error to parameterise the model.

Examining the observed soil moisture one can see the loss of water from the bottom of the soil column due to the pumping. Another obvious feature is the variability in soil moisture down

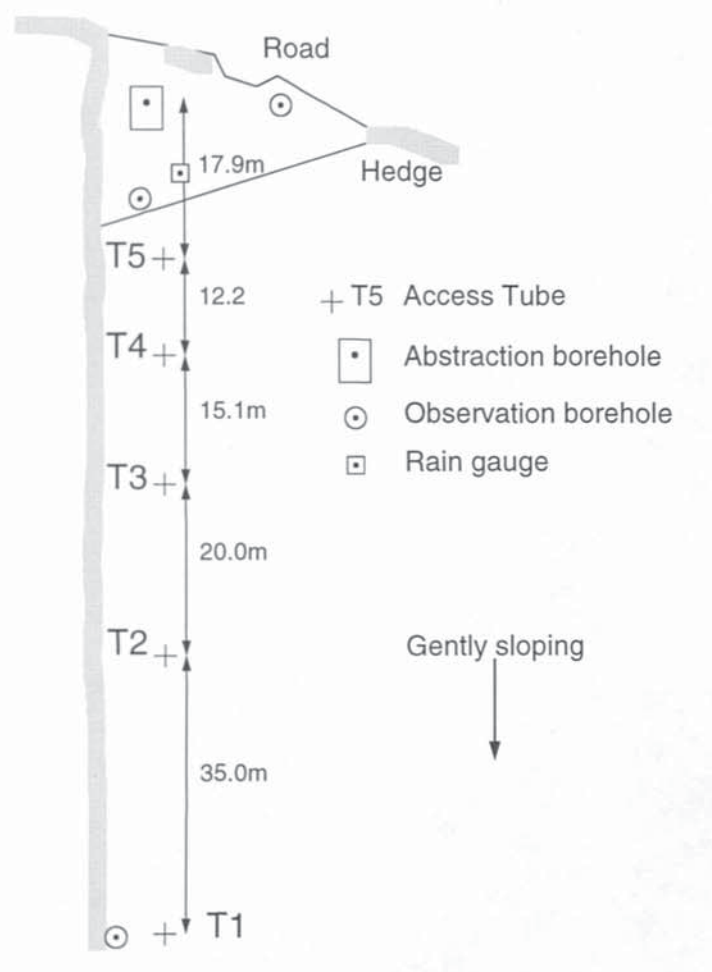


Figure 6.16: Plan of Heath House site.

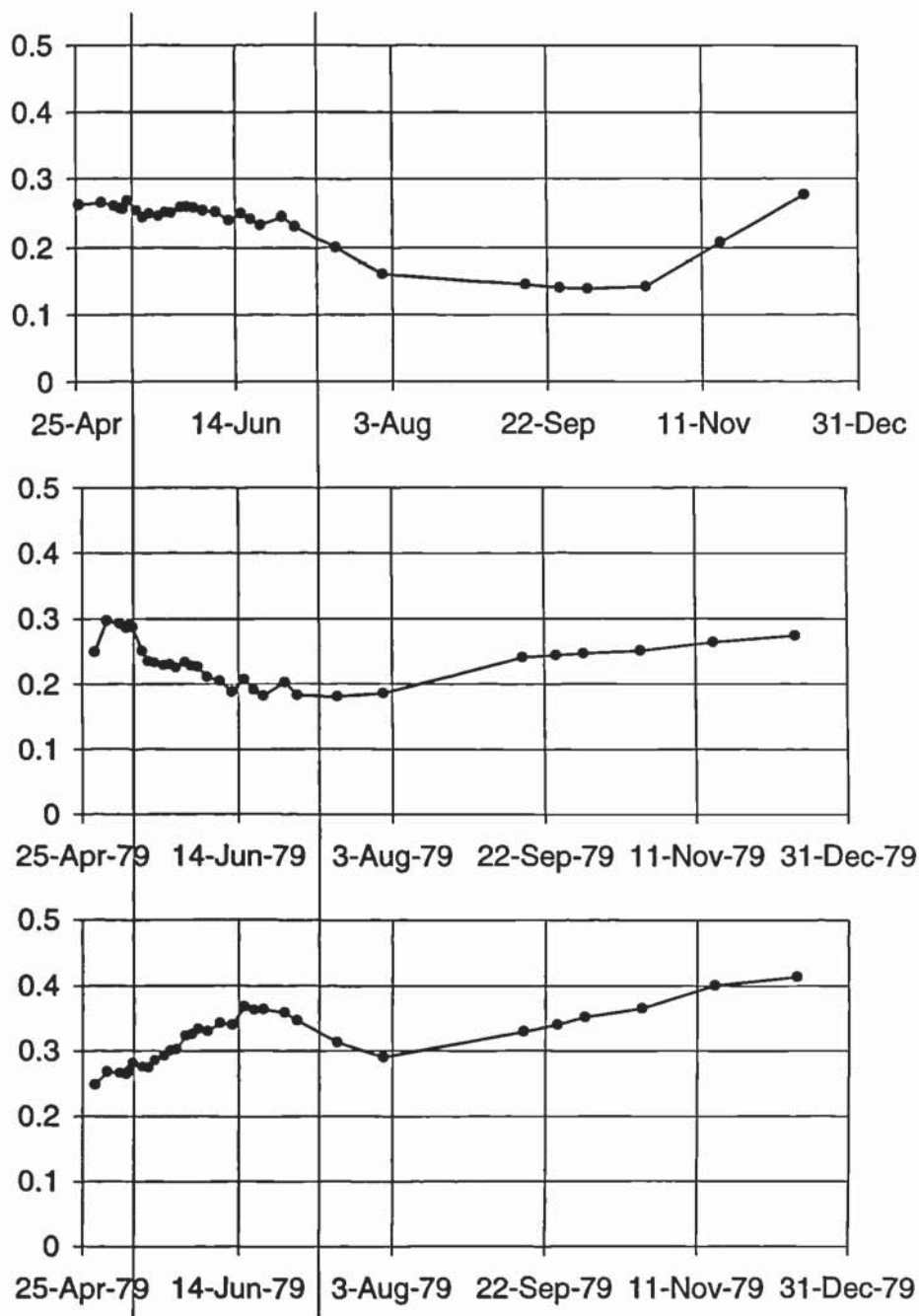


Figure 6.17: Development of soil moisture in the root zone; (a) the observed soil moisture, (b) modelled using trial and error to generate input parameters and (c) modelled using regression equations to generate parameters.

the soil column. Though the soil moisture changes through time many of the features, the peaks and the troughs, remain throughout the period. This variability can thus be put down to soil properties, in particularly those properties that influence how the soil retains moisture against applied tension, rather than to transient movement of water.

The difference between the observed and the either of the models is clear. Generally the models present a much smoother soil moisture profile. The models were both initiated with the observed soil moisture profile. Within the first few days of the model the profile, particularly of the trial and error, model has considerable become much smoother. This is a similar process to the movement and smoothing of the of the groundwater surface observed in the regional model in Section 5.8.1. In both the soil moisture models the loss of moisture at the bottom in response to the lowering of the watertable is apparent.

In terms of capturing the broad response of the loss of moisture from the profile due to the lowering of the watertable both models have been successful. In capturing the detail of the soil moisture profile, and thus justifying the use of such a complex model compared with simpler layer models they have been less successful.

6.7 Discussion

The soil moisture modelling exercise reported above provides much useful information about the problems that would be encountered when using the model across the entire region as envisaged in Figure 3.11. The principal consideration has proved to be whether the data, which would be of lower quality than available at the Heath House, would be adequate for such an exercise.

In the previous chapter the use of geostatistics was discussed (Chapter 5, Section 5.7). Such methods could equally easily be applied to soil properties, rather than revise that work a set of techniques based on a similar but slightly different paradigm was explored and is reported in this discussion these are techniques that use stochastic assumptions.

Other considerations that have arisen from the modelling exercise are the use of regression equations for parameterising models and the problems of calibration.

6.7.1 Modelling with lower quality data

The soil at the Heath House site and around access tube 5 has been well documented at a large scale (e.g. the pits that Hedges dug) however we can hardly expect this to be the case over the entire region of interest in any similar modelling effort. Access tube 5 is in a region designated as being a soil of the Newport soil series. This series is a *brown earth* and described in Crompton and Osmond (1954) as ranging between loamy sand, sandy loam and sand. The Newport Series was first fully described by Jones (1975). Is this information adequate for modelling?

The preliminary results from the modelling above appear to indicate that modelling using general parameters from tables and the literature can be as successful than modelling based on

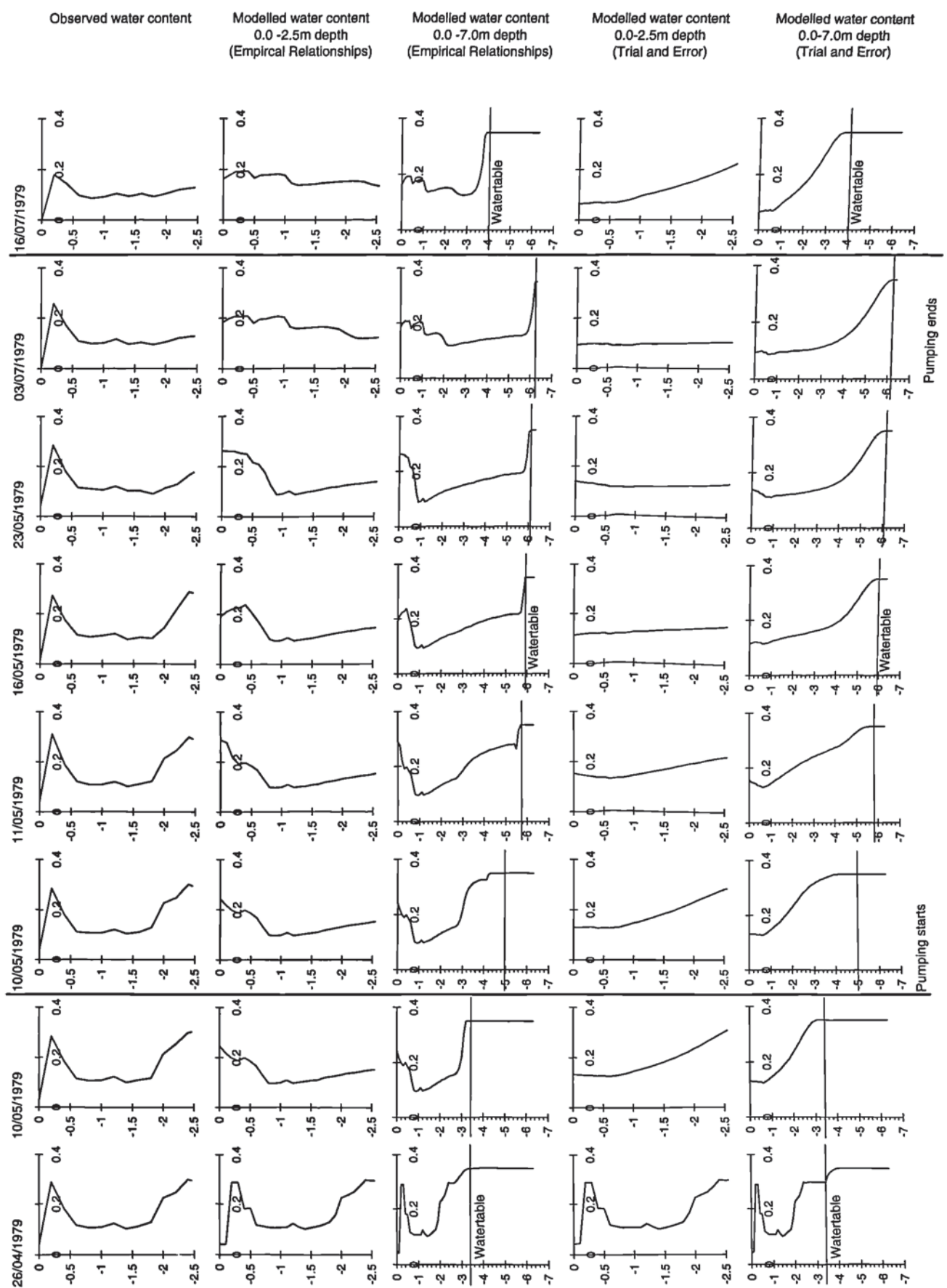


Figure 6.18: Observed and modelled soil moisture profiles, if viewed from the side then each column corresponds to one observation time. The top row shows the observed soil moisture profile in the top 2.50m, the second and third rows show the modelled soil profile using the regression equations in the top 2.50m and 7.00m and the forth and fifth rows similarly show the modelled profile using trial and error and standard values.

site data and regression equations derived by other people. However, neither approach generated a satisfactory simulation of the soil moisture regime.

One problem is that it may not be the characteristics of the horizon that are important but the characteristics of the soil horizon boundaries. In the observation above (the first row in Figure 6.18) there is a persistent peak in the moisture content profile at the same depth (25cm). This effect is mimicked to some extent in the first model, which seems to spread this effect out over a band, but it is completely missing in the second model. One might speculate that the peak is caused by properties of a localised transition zone rather than of a whole horizon, e.g. the depth of ploughing. When Hedges sampled the soil horizon, to a certain extent he captured some of the soil properties that are causing this peak but the coarseness of the sampling interval caused its specific characteristics to be mixed in with and to be spread over a whole horizon.

Of course an alternative explanation is that this peak exists in a gap between the large evaporation at the soil surface and a zone in which the sugarbeet roots remove water in the largest quantities. Without more precise knowledge of the sugar-beet's behaviour and the soil properties this cannot be resolved. Further work is needed to explain this.

Another problem is finding an initial starting moisture content profile from which to begin the simulation. One solution is to begin from saturated conditions and allow the soil to drain.

6.7.2 Generating maps of stochastic variables

It was the aim to apply the soil moisture model over a region rather than at a well documented site. As discussed above under these conditions the quality of data would be greatly reduced. In the Shropshire area one would have to use the soil maps and their associated memoirs (e.g. Crompton and Osmond, 1954).

A soil series basically segments a region into zones of broadly similar soil type, however it does not capture all spatial variability. Probably the only way of dealing with this variability within soil series is by using some stochastic assumptions. There was no time to explore this idea fully but Charnock, Elgy and Hedges (1996) presented a simple procedure for generating realisations of spatially varying soil properties that could be implemented using cartographic modelling functions¹. This would have been very useful for generating realisations of stochastically varying input parameters. Unfortunately the failure of the soil moisture model meant that regional modelling could not be performed, but the method of generating the maps is presented because it is a useful technique, and could have application elsewhere.

Overview of stochastic simulation

Stochastic simulation is the process of generating an artificial set of data with similar statistical properties to a real set of data. It is a technique that has been applied to non-spatial and

¹ The credit for the mathematical derivation should go to John Elgy, Dept Civil Engineering, Aston University

temporal phenomena and to 1, 2, 3 and 4-dimensional spatio-temporal phenomena. The product of stochastic simulation algorithm is known as a realisation as it is just one possibility from a large, possibly infinite, number of configurations that could be true. Generally a number of realisations are generated, to give an indication of the variability that might be expected in the environment or to be used in a Monte-Carlo analysis of uncertainty.

When designing an algorithm simplifying assumptions about the stochastic properties of the phenomenon are made. The algorithm will not be able to simulate all the statistical properties but, if it is well designed it will simulate those that are important. Thus when simulating spatial data as well as reproducing the mean and variance it would be appropriate to simulate properties of spatial autocorrelation as well. There are many algorithms in the literature and the reader is referred to Deutsch and Journel (1992) and Dowd (1994) for a discussion of these.

In Section 5.7.1 the techniques and principles of geostatistics were briefly outlined, and it is useful here to compare classic geostatistics as typified by ordinary kriging and stochastic techniques. As stated before kriging is the “best linear unbiased estimator”, i.e. the variance of the residuals is minimised, but in a stochastic realisation this is not the case. So what is the advantage of a stochastic simulation? In Section 5.7.1 Isaaks and Srivastava’s (1989) criteria for a good estimator were given. criterion 1 stated that the estimator should produce an estimation with a distribution with similar properties to the true distribution. Kriging techniques smooth the phenomenon and thus, as far as measure of spread are concerned, they do not satisfy this criteria. The semivariogram generated from a kriging exercise will have a different form to the one generated from the original observed data, generally it will have a longer range. Stochastic simulations do satisfy this criteria. The choice of whether to use one or the other will depend on the task. Where the accuracy of an estimate at any given point is important the kriging should be used. Where the influence of spatial variation is important an analysis that incorporates a stochastic simulation technique should be used.

To generate a stochastic realisation information is required about the statistical properties of the phenomenon. Sometimes these can be inferred from knowledge of the physical process that generate the phenomenon, but also they can be generated from observations. When generating a realisation from observations it is desirable that the realisation should conform to the observations. This is known as a conditional realisation. The realisation is unconditional when the value generated by the algorithm and the value actually observed at a point do not necessarily match. Often an unconditional realisation is modified to a conditional realisation by applying a second stage of processing, typically simulated annealing. Briefly, in simulated annealing an energy function is defined for the phenomenon that is at its lowest when the realisation is conditional. The realisation is randomly changed, if the change results in a lower energy function then it is accepted otherwise it is rejected.

An assumption that is frequently made in stochastic simulation algorithms is that of stationarity, i.e. the location of the mean and magnitude of the variance do not drift with spatial location.

This clearly does not hold in all cases. One way of accounting for non-stationarity is the “reproduction of major heterogeneities” (Deutsch and Journel, 1992). This idea is based on the assumption that the most significant changes in the underlying distribution occur abruptly at boundaries. For example, the drift in porosity within a sedimentary rock unit caused, say, by a gradual trend to small sized particles having been laid down during sedimentation, might be considered insignificant to the change in porosity that occurs at the boundary between the sedimentary and an igneous unit.

There are two approaches to delineating major heterogeneities. The first is to generate them with a stochastic algorithm. The whole process is then two stage, the first is to stochastically generate the units and the second is to generate the stochastic realisation within the units. Alternatively the position and dimensions of the units may be already known, and the stochastic realisation is generated within these prior mapped units.

There can be a degree of dependence between spatial variables. For example concentrations of different deposited pollutants might exhibit considerable correlation. Theoretically some cross-correlation matrix could be established which would enable correlated realisations of different variables to be generated. In practice an approximation is often used. One of the simplest approximations is to generate one variable initially and then to generate the correlated variables stochastically from that realisation.

This has been a necessarily brief introduction to stochastic simulation, which is a large and expanding subject. The next section describes a stochastic algorithm which puts some of this theory into practice.

A simple cartographic modelling approach to generating stochastic realisations

The underlying principle of the technique is that an assumption of a stationarity of a soil property within a soil series can be made. That is to say the mean and variance of a property, for example, saturated conductivity do not vary. Major heterogeneities are therefore accounted for by prior mapping of the soil units. The procedure produces an unconditional realisation.

The procedure is as follows; assume a spatial autocorrelation function that is isotropic and if the correlation between two cells $1m$ apart is φ then the correlation of between two cells n -metres apart is φ^n . Standardise the property to zero mean and unit standard deviation by:

$$v_t = \frac{f - \bar{f}}{s} \quad (6.21)$$

where:

- v_t — is a standardised spatial property;
- f — the spatial property under consideration;
- \bar{f} — the mean of the spatial property;
- s — standard deviation of the spatial property.

The model is then:

$$v_{[i,j]} = av_{[i-1,j]} + av_{[i,j-1]} + b\eta_{[i,j]} \quad (6.22)$$

where:

a & b are constants to be estimated;

η a random number of mean 0 and standard deviation 1.

Multiplying Equation 6.22 by $v_{[i,j]}$ and taking the expectations:

$$\begin{aligned} E(v_{[i,j]}v_{[i,j]}) &= aE(v_{[i-1,j]}v_{[i,j]}) + aE(v_{[i,j-1]}v_{[i,j]}) \\ &\quad + abE(v_{[i-1,j]}\eta_{[i,j]}) + abE(v_{[i,j-1]}\eta_{[i,j]}) + b^2E(\eta_{[i,j]}\eta_{[i,j]}) \end{aligned} \quad (6.23)$$

and knowing that:

$$\begin{aligned} E(v_{[i,j]}v_{[i,j]}) &= 1 \\ E(v_{[i-1,j]}v_{[i,j]}) &= E(v_{[i,j-1]}v_{[i,j]}) = \varphi^n \\ E(v_{[i-1,j]}\eta_{[i,j]}) &= E(v_{[i,j-1]}\eta_{[i,j]}) = 0 \\ E(\eta_{[i,j]}\eta_{[i,j]}) &= 1 \end{aligned}$$

it follows that:

$$a\varphi^n + a\varphi^n + b^2 = 1 \quad (6.24)$$

multiplying Equation 6.22 by $v_{[i-1,j]}$ and taking expectations:

$$\begin{aligned} E(v_{[i,j]}v_{[i-1,j]}) &= aE(v_{[i-1,j]}v_{[i-1,j]}) + aE(v_{[i,j-1]}v_{[i-1,j]}) \\ &\quad + bE(v_{[i-1,j]}\eta_{[i,j]}) \end{aligned} \quad (6.25)$$

and knowing that:

$$\begin{aligned} E(v_{[i-1,j]}\eta_{[i,j]}) &= 0 \\ E(v_{[i,j]}v_{[i-1,j]}) &= \varphi^n \\ E(v_{[i-1,j]}v_{[i-1,j]}) &= 1 \\ E(v_{[i,j-1]}v_{[i-1,j]}) &= \varphi^{\sqrt{2}n} \end{aligned}$$

then:

$$\varphi^n = a + a\varphi^{\sqrt{2}n} + 0 \quad (6.26)$$

from Equation 6.26:

$$a = \frac{\varphi^n}{1 + \varphi^{\sqrt{2}n}} \quad (6.27)$$

and from Equation 6.24

$$b = \sqrt{1 - 2a\varphi^n} \quad (6.28)$$

Equation 6.22 is a simple formulation and can be implemented within a map algebra facility (i.e. *r.mapcalc*). In order to do this a new function was added to *r.mapcalc* in order to generate the normalised random variable, Appendix A.3.3. Figure 6.19 gives an example of the output that can be produced.

Rasters of other soil properties can be generated in a similar fashion or by assuming some correlation between variables. One is left with the task of assigning means and standard deviation to the soil series at different horizons, fortunately Soil Survey of England and Wales (1982) give a list of just such values gathered from the available literature for the main horizons of many of the soil series within the Tern area.

6.7.3 The use of regression equations

The use of regression relationships to determine input parameters has to be questioned, as it has not proved very successful for this project. As an example take the *alpha* parameter (in Equation 6.18). In the Table 6.3 the fourth layer has a very large percentage of gravel. This is a very unusual proportion and can be expected to have a significant affect on the hydrologic properties of the soil. However the regression equations (Equation 6.14 to 6.17) do not explicitly include percentage gravel as an explanatory variable. If Evans's (1992) *alpha* equation (Equation 6.16) is applied to the data uncritically, the *alpha* value for layer 4 is considerably lower than the other layers (Table 6.4).

This is intuitively wrong, as if *alpha* is taken as the inverse of the air-entry point (as is suggested in by Evans's (1992) Equation 6.12), then one would expect a gravely soil to have an air-entry point very close to zero and thus a very high *alpha*.

6.7.4 Calibration

The calibration of the soil moisture component has proved a very problematic part of this component of the project, and it is likely that over a larger area this will prove even more so. Whilst each point on the landscape is to be modelled independently (i.e. lateral flow is not considered);

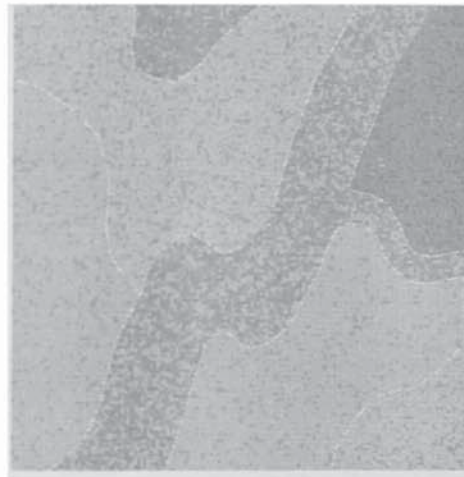
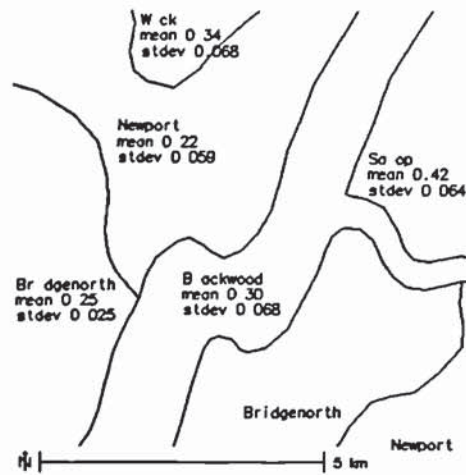


Figure 6.19. (a) a map of soil series (made-up because of copyright) and (b) a stochastic realisation of the variation of saturated moisture content. There is of course some correlation over the boundaries of soil series, but this is not an unrealistic proposition. This raster would then be used as one of the layers in Figure 6.12.

layer	Evans' α
1	68.08
2	57.43
3	29.30
4	6.32
5	12.20

Table 6.4: Calculated value of Evans's (1992) α .

for theoretical reasons (i.e. soil properties are spatially correlated) and for practical reasons, (i.e. the soil moisture profile is being modelled at hundreds if not thousands of points across the landscape) the input parameters are generated as spatially varying rasters. Thus calibration at each individual point is not a possibility.

The documented soil moisture sites of the Shropshire Groundwater Scheme give information against which the model can be calibrated, though this has proved tricky and more work needs to be done. The simplest approach to calibration is to assume that the sites are representative of a wide area and use the calibrated parameters. The alternative is to modify the raster generating functions (e.g. the spatial interpolation of rainfall) and to assume that when the parameters generated at the documented sites give good results then parameters at undocumented sites are also being well specified. Obviously this can be a circular argument if one is generating the rasters (through interpolation by whatever method) based on data at specific sites and then comparing the results against the observed soil moisture development at the same sites.

Hopmans and Gutierrez-Rave (1988) discussed the problem of calibrating the root uptake function of the soil moisture model SWATRE² over a large area with a spatially varying soil (the root uptake function in SWMS_2D is based on the SWATRE function). They used a Monte-Carlo analysis with a trial and error method to optimise parameters, and this gave better transpiration results when compared with single site calibration. However, they found a considerable error between measured and simulated transpiration in particularly dry years.

By fitting part of the output to an easily observable spatially varying measure, can the assumption be made that other parts of the model output are of a similar accuracy? For example, if the moisture content of the top few centimetres is well modelled does this mean that another soil moisture content deeper into the soil is also well modelled? If this is a reasonable assumption then microwave remotely sensed data could provide a way of spatially calibrating the model. The use of microwave remote sensing for soil moisture monitoring is a current research topic, see for example Rogowski and Engman (1996). Compared with calibrating to limited site data for a few points this would reduce the variability of uncertainty across the landscape.

The top few centimetres are largely controlled by atmospheric conditions (see the profiles, in Figure 6.18, of soil moisture content to see how the behaviour of the moisture in the top of the soil profile bears little resemblance to the behaviour further down³), and the effect we are most interested in is the change in soil moisture in response to watertable movement and so, for this project, this would not be an appropriate measure against which to calibrate.

The approach to calibration adopted here has been to model soil moisture at a site, to vary and degrade the inputs and discover how the model responds to this. The assumption is that the lessons learnt during this stage are valid across the landscape at different sights. Regression

² SWATRE is a later version of SWATR, the root sink function was discussed in Section 6.1.4.

³ It should be noted that the difference in the surface soil moisture could be an artefact of the method of measurement, i.e. with a neutron probe at the surface many of the neutrons will be emitted into the atmosphere and not reflected back to the sensor.

equations by other authors have proved inadequate for parameterising the model, and work needs to be done to resolve this problem.

6.8 Recommendations

In summary several recommendations can be drawn from the experience.

- Use a much simpler model, in retrospect the model of van Keulen and van Laar (1986) (see Section 7.2.5) would have been much more appropriate.
- Tables of standard values are extremely useful, and are not prone to giving extremely unrealistic values in the way wrongly used regression equations are.
- Writing a clever model is not really enough without good documentation, quality assured sample input files (together with output files), tables of standard values, and possibly a procedure for constructing the model is also needed.

Chapter 7

Crop modelling

Plants and crops are modelled over a huge range of scales from cellular interactions within a single plant organ to yield on a continental or global scale. Section 7.1 in this chapter will review the essential science of crop growth. With a discussion of crop models and crop modelling in Section 7.2 Section 7.3 gives examples of the ways GIS technology has impacted on crop modelling and might impact in future. Remote sensing of crop distribution is discussed in Section 7.4. Crop modelling was the last component of the overall environmental system to be examined, the other two components being of course the groundwater and soil moisture (Chapters 5 and 6). It is thus the least well developed of the three, however, a simple crop model was implemented using *rmapcalc* and though there was no time to use it on the Tern valley, the practical issues are discussed in Section 7.5. On the basis of the literature review, the remote sensing investigation and the practical lessons learnt from repeating Hedges's (1989) analysis (Section 4.5) and from implementing a crop model, recommendations are made in Section 7.6.

7.1 Basics of plant growth

The literature on plant physiology, crop growth and crop yield is extensive, see for example van Keulen and Wolf 1986, and only the basic principles will be reviewed here.

Plants use energy from sunlight to convert carbon dioxide (CO_2) to carbohydrates. This process is called photosynthesis and occurs in the plant leaves. The carbohydrates generated are converted to plant material or expended in respiration. For any particular plant there is therefore a maximum assimilation rate of CO_2 which depends on the incident radiation and on the efficiency of the plant. The major limitations on incident radiation are latitude, aspect, cloudiness and shading by landscape features or by other plants.

CO_2 enters the leaves through stomata – small holes in the leaf skin. Through these stomata the plant loses moisture, this is transpiration. The plant takes in moisture through the roots, and there is thus a flux of water from the soil, through the roots, the stem and out through the leaves

into the atmosphere (see van Keulen and van Laar, 1986; Barry, 1976). Many plants are able to adjust their stomata to control transpiration when soil water is limited, this also has the effect of reducing the rate of CO_2 assimilation and reducing growth rate.

The term potential transpiration is used to describe the transpiration from a crop that is adequately supplied with water. Under these conditions it is meteorological factors, such as atmospheric demand for water, which control transpiration up to the time when physiological features of the plant act as a constraint. The term actual transpiration is the transpiration from a plant under the actual existing soil moisture conditions. If water is freely available this will, of course, be the same as potential transpiration.

In practice it is common to consider both actual and potential *evapotranspiration* — terms which include the direct evaporation of water from the soil in addition to the moisture transpired from the crop. The reason for this is that it is hard to separate the various hydrological components which contribute to evapotranspiration, though frequently a surrogate such as leaf area index (LAI

the ratio of leaf area to ground area) is used to partition the total evapotranspiration between the soil and the crop.

Plants need nutrients, elements and compounds, obtained from the soil, to facilitate various internal metabolic functions. When these are limited, growth and yield are also constrained. Many crop models include the consideration of nutrients, but for this project nutrient limited conditions are not considered

7.2 Crop modelling

Crop models come in a variety of scales and complexity, from those that are very detailed and attempt to simulate in-plant fluxes of energy and materials and physiological changes due to environmental conditions, to those that operate on continental scale and are used to assess the potential production of nations. Only those models concerned with crop yield prediction are dealt with here, with a bias to the simple and those that operate on a field scale. As Hanks and Hill (1980) point out these models:

“ ... of necessity, use simplifying assumptions to replace details of plant responses to the environment with less complex relationships. ”

For example it is fair to say that most models simulate yield by relating estimates of transpiration to material assimilation, because the movement of water vapour from the leaf is proportional to the movement of CO_2 into the leaf. As has been noted, under conditions of adequate water supply a plant will transpire at the potential rate, and the cumulative assimilation and respiration are thus proportional to the cumulative potential transpiration. As water becomes limited the plant reduces its transpiration by closing its stomata, so the movement of CO_2 into the leaf will reduce proportionally to the reduction in transpiration, thus assimilation and respiration are proportion to actual transpiration.

The mechanism by which plants detect the soil moisture deficit in the soil and transmit this information to the leaves is contentious, the two main camps argue either for hydraulic or for hormonal signal propagation. That the information is transmitted is not contentious, and it is usual to assume a simple relationship between soil moisture and transpiration.

More sophisticated models simulate moisture being taken up by the roots. To this extent they often overlap with the soil models discussed in the previous section (6.1.2), and model packages tend to include both plant and soil models within a single framework, e.g. Feddes et al. (1978) SWATR and CROPR. However, these models still use estimates of actual and potential evapotranspiration, as the means of estimating production.

7.2.1 Potential and actual evapotranspiration

Various methods are available for measuring actual and potential evaporation and transpiration, see Shaw (1993) for a discussion of these. Measurements of transpiration and evaporation can be used for constructing tables of typical values of these variables at a particular site or over a constrained region. More common practice is to measure basic meteorological variables and use various published equations to convert these into potential evaporation and transpiration.

Potential evapotranspiration

Two distinct approaches for estimating potential evaporation and evapotranspiration can be identified. The first approach involves mass transfer, and uses the vapour pressure gradient between the evaporating surface (normally taken as saturated vapour pressure) and the vapour pressure of the receiving environment. A function of wind speed is assumed to assist the movement of water down this gradient. Shaw (1993) gives the formula as:

$$E = f(u)e_s - e_a \quad (7.1)$$

where

- E evaporation rate (LT^{-1});
- u wind speed at a certain height, commonly 2m (LT^{-1});
- e_s saturated vapour pressure ($MT^{-2}L^{-1}$);
- e_a vapour pressure of the air.

The second approach is the energy budget method. A certain amount of energy is required to evaporate water and this energy comes from the sun, Shaw (1993) gives the amount of energy available for evaporation of a unit area column of water as:

$$Q_E = Q_s - Q_{rs} - Q_l - Q_c \pm Q_g \pm Q_v \quad (7.2)$$

where:

- Q_E energy available for evaporation;
- Q_s incoming short range solar radiation;
- Q_{rs} reflected short range solar radiation;
- Q_l emitted long wave radiation;
- Q_c sensible heat transfer to the air;
- Q_g the change in stored energy;
- Q_w the energy transfer between the water and the bed.

Thus the amount evaporated from the open water body E_o is

$$E_o = \frac{Q_E}{\lambda_p} \quad (7.3)$$

where

- λ the latent heat of vaporisation of water.

The most widely used practical method of estimating potential evaporation is the Penman equation see Shaw, 1993 This is termed a combination method because it uses a combination of the two above approaches Potential evapotranspiration (PE) is modelled by a function of available radiant energy and a term combining atmospheric moisture deficit and wind speed, Equation 7.4 Penman's aim was to produce an equation based on variables that are easy to measure or obtain from tables Over the years various modifications have been made to the equation which have extended its application from potential evaporation from a water body to potential evapotranspiration

$$PE = \frac{\frac{\Delta}{\gamma} H + E_a}{\frac{\Delta}{\gamma} + 1} \quad (7.4)$$

H is the available radiant energy and is usually obtained from tables of radiation fixed by latitude and season and then applying the equation:

$$H = R_i(1 - r) - R_o \quad (7.5)$$

where

- r the albedo (the proportion of incoming radiation is reflected): albedo for short cut grass is usually taken as 0.25,
- R_i incoming radiant energy at a given latitude and season;
- R_o out going radiant energy.

E_{at} is a term for atmospheric moisture deficit:

$$E_{at} = f(u)(e_a - e_d) \quad (7.6)$$

where

- $f(u)$ a function of wind speed at a given height, e.g. 2m;
- e_a saturated vapour pressure at prevailing temperature;
- e_d the vapour pressure of the air.

In Equation 7.4 γ is known as the hygrometric constant, and Δ is the slope of the curve of saturated vapour pressure against temperature at the prevailing temperature.

Actual evapotranspiration

Another important parameter in crop models is the actual evaporation AE . The Penman-Monteith Equation is a widely used method for its calculation:

$$AE = \frac{\Delta R_n + \frac{\rho c_p (e_a - e_d)}{r_a}}{\lambda \left(\Delta + \frac{\gamma (r_s + r_a)}{r_a} \right)} \quad (7.7)$$

where

- AE actual evapotranspiration ($gm^{-2}s^{-1}$);
- R_n net radiation ($W m^{-2}$),
- ρ density of air,
- c_p specific heat of air,
- r_s net resistance to diffusion through leaves and soil surface (sm^{-1});
- r_a net resistance to diffusion through the air (sm^{-1}).

The Penman Monteith expression Equation 7.7 is just one of many expressions for calculating EA. Evapotranspiration can have considerable spatial variability, and frequently recourse is made to the many empirical equations, though these tend to be specific to a certain region. Dugas and Ainsworth (1985) looked at how different methods of estimating potential evapotranspiration affect the output of different crop yield models. They found significant and consistent difference when the Penman equation was used compared with another commonly used method, the Priestley-Taylor equation.

7.2.2 Simple crop models

Stewart and Hagan (1973) proposed an expression for relative yield based on actual and potential evapotranspiration

$$\frac{Y}{Y_M} = 1 - \beta_0 + \beta_0 \frac{AE_s}{PE_s} \quad (7.8)$$

where

- Y dry matter yield ML^{-2} ,
- Y_M maximum dry matter yield ML^{-2} ,
- β_0 the slope of $\frac{Y}{Y_M}$ against $\frac{AE_s}{PE_s}$ curve;
- AE_s seasonal actual evapotranspiration,
- PE_s seasonal potential evapotranspiration

Stewart and Hagan 1973 used evapotranspiration which combines soil evaporation with transpiration from the plant. As Hanks and Hill 1980 noted.

" Models relating yield to transpiration are more sound than those relating yield to evapotranspiration because they account for water that goes through the plant. "

But as, Hanks and Hill also pointed out, it is hard to separate transpiration from evaporation, so Stewart and Hagan's 1973 approach can be seen as a pragmatic decision to use less closely related but more easily obtainable parameters. Jensen suggested the model:

$$\frac{Y}{Y_M} = \left(\frac{T_1}{T_{M1}} \right)^{\lambda_1} \cdot \left(\frac{T_2}{T_{M2}} \right)^{\lambda_2} \cdot \dots \cdot \left(\frac{T_n}{T_{Mn}} \right)^{\lambda_n} \quad (7.9)$$

where

- T_i transpiration for growth stage i
- T_{Mi} maximum transpiration for growth stage i
- λ_i weighting factor for growth stage i

Hanks and Hill 1980 used this model and modified transpiration using the equation

$$\begin{aligned} T &= \frac{T_M}{b} \frac{SWC}{AW} & \text{for } \frac{SWC}{AW} < b \\ T &= T_M & \text{for } \frac{SWC}{AW} \geq b \end{aligned} \quad (7.10)$$

where

- b ratio of $\frac{SWC}{AW}$ below which transpiration is limited;
- SWC soil water content,
- AW available soil water

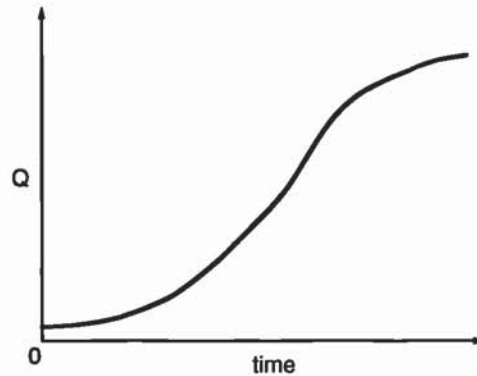


Figure 7.1: Typical “S” shaped cumulative crop production curve. Q is the cumulative yield (ML^2).

7.2.3 Regression models

An alternative approach is the use of regression models such as those developed and used by Hedges (1989) and which have been described in Section 2.2.7. To recap Hedges generated expressions that related moisture debt (the loss of moisture in the root zone \times the period of moisture debt) to yield decline from the potential yield. de Wit and Penning de Vries (1985) said of regression models:

“...they remain very specific and their refinement has not led to major improvements in accuracy or generality. ”

Hedges’s (1989) models can be justified because they were developed from data from crop experiments conducted in the UK in soil classified as belonging to the Newport soil series — a dominant soil series in Shropshire. They could be criticised in that they were not developed specifically from crop experiments in Shropshire.

7.2.4 CROPR

Feddes et al. (1978) published a crop model called CROPR, which was designed to work with the soil moisture model SWATR. Over the years these models have evolved into a commercial product SWACROP. Feddes et al. noted that cumulative yield for an annual crop can be expected to resemble an “S” shaped curve (Figure 7.1). Feddes et al. expressed this as Equation 7.11.

$$\frac{dQ}{dt} = f(Q.t) \quad (7.11)$$

where:

Q — cumulative yield per unit area (ML^{-2});

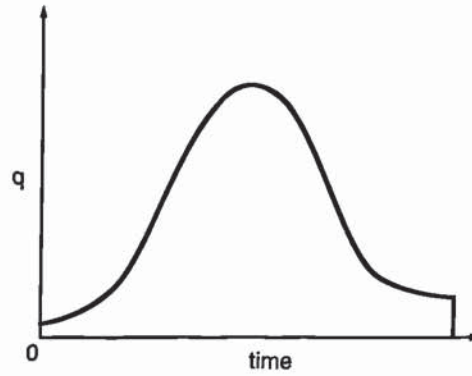


Figure 7.2: Typical variation of growth rate through a season.

t — time;

they used a daily approximation of growth rate (Equation 7.12), with Figure 7.2 showing how it can be expected to change during a growing season;

$$\dot{q} = \frac{\Delta Q}{\Delta t} \approx \frac{dQ}{dt} \quad (7.12)$$

where:

\dot{q} — approximate growth rate ($kg.ha^{-1}.day^{-1}$);

The actual growth rate, \dot{q}_{act}^i is obtained from the potential rate \dot{q}_{pot} using Equation 7.13;

$$\dot{q}_{act}^i = \frac{A}{2}w^i + \frac{\dot{q}_{pot}^i}{2} - \frac{1}{2}[(\dot{q}_{pot}^i + Aw^i)^2 - 4\dot{q}_{pot}^i Aw^i(1 - \xi)]^{\frac{1}{2}} \quad (7.13)$$

where:

\dot{q}_{act}^i — the actual growth rate on day i ($kg.ha^{-1}.day^{-1}$);

\dot{q}_{pot} — potential growth rate ($kg.ha^{-1}.day^{-1}$);

A — slope of the growth rate against water growth factor graph ($kg.ha^{-1}.mm^{-1}$);

w — water supply factor (mm);

ξ — an empirical constant close to zero.

The exact derivation of this equation can be found in Feddes et al. (1978), but in effect on any day there are two asymptotes which confine the growth rate, Figure 7.3. The sloping asymptote is the limit imposed by the supply of water, the horizontal asymptote is the limit imposed by the physiology of the plant and the maximum incident radiation. As with the simpler models plant transpiration is a crucial factor, and Feddes et al. use the relationship:

Figure 7.3: Feddes et al.'s (1978) asymptotes of growth rate, when $\xi = 0$. This graph will be different for each day as potential growth rate changes.

$$w^i = \frac{E_{pt}^i}{\Delta e^i} \quad (7.14)$$

where:

E_{pt}^i plant transpiration on day i ($mm \cdot day^{-1}$);
 Δe vapour pressure gradient (mbar).

w^i is proportional to the plant transpiration, at a given vapour pressure (i.e. the CO_2 gradient is considered constant), and the actual plant transpiration must therefore be calculated to enable Equation 7.13 to be applied. This will depend on the soil moisture status, taken from the SWATR component of the model. A higher vapour pressure gradient will force a higher plant transpiration, but not an increase in growth rate (hence it is included in Equation 7.14 as a denominator).

Like many models CROPR depends on a good estimate of potential growth rate on a particular day. Feddes et al. use the equation:

$$\dot{q}_{pot} = [\Lambda P_o + (1 - \Lambda) P_c] \cdot \phi_r \cdot \alpha_T \cdot S_c \cdot \beta_h \quad (7.15)$$

where:

\dot{q}_{pot} — potential growth rate ($kg \cdot ha^{-1} \cdot day^{-1}$);
 Λ fraction of day overcast;

Figure 7.4 The CROPR model, after Feddes et al. (1978)

- P gross growth rate on an overcast day $kg\ ha^{-1}day^{-1}$;
- P gross growth rate on an cloudless day $kg\ ha^{-1}day^{-1}$;
- r A respiration factor
- α_T photosynthesis efficiency temperature function;
- S soil coverage by the crop,
- λ factor to partition photosynthesis between harvested parts and unharvested parts.

P and P can be read from tables and depend on latitude and date. α_T is included to account for changes in photosynthetic efficiency at different temperatures (again there are tables for this).

The final yield over a season Q_{act} in $kg\ ha^{-1}$ is given by:

$$Q_{act} = \sum_{i=1}^n q_{act} \Delta t \quad (7.16)$$

A string of equations is a very terse way of communicating a model, and so Feddes et al. summarised the main principals of their model with a flow diagram, reproduced in Figure 7.4.

7.2.5 van Keulen's (1986) crop model

As with Feddes et al.'s (1978) model, the model presented by van Keulen (1986) also follows the sequence from potential production to crop production limited by actual conditions. The book, *Modelling of Agricultural Production Weather Soil and Crops* (van Keulen and Wolf, 1986), includes work from a number of contributors. Throughout the book the complexity of the model is incrementally increased from a simple potential production model to a nutrient and moisture



Figure 7.5: van Keulen and van Laar's (1986) crop model.

limited actual production model. The intention was that this model could be used with a pocket calculator, though a FORTRAN implementation of the model was also given.

In chapter 3.4 van Keulen (1986) presents a refinement of the model limited only by moisture. It uses a similar underlying crop model as CROPR, but where CROPR links into the soil moisture model SWATR to simulate the soil moisture regime, van Keulen's (1986) model uses a much more conceptual (in the hydrological sense, Section 3.2) and simple model of soil moisture.

The time-dimension is divided into steps of about 10 days. van Keulen's (1986) model uses a concept of degree days to track the growth of the plant through the distinct development stages of its growth. A crop that has experienced 10 days at 12°C accrues 120 degree days ($10 \times 12 = 120^{\circ}\text{C}$). In this model a plant must collect a crop specific number of degree days to complete each growth stage.

The vertical dimension is divided into two layers of variable depth, one is the root zone the other is the potential root zone. At each time-step during the model's operation, a water balance estimation is performed on the root zone soil. Inputs to the system include irrigation and precipitation (boundary conditions specified by the model user), capillary rise from a shallow water table (watertable depth is specified by the user and capillary rise for different soil types is given by tables) and additional moisture reserves tapped as roots grow downwards. Outputs are evaporation from the soil, transpiration from the crop and drainage from the root layer, Figure 7.5.

At each time-step the gross assimilation is divided between respiration (based on dry weight) and increase in plant organs. The partition of the carbohydrates available for plant organ growth

is based on partitioning factors which depend on the development stage — these are read from tables. In fact much of the data required for the model is generated from tables given in van Keulen and Wolf (1986) either using a soil type or a crop type as a key.

There are many other models in the literature conceptually very similar to the ones described above. At this scale crop models can be said to be mature and much of the current work is to make models accessible to the agricultural decision making community. In the past this has meant making models that are achievable with a calculator. More recently as computers have become more widely available, the emphasis has changed to embedding such models within helpful user interfaces. This is described in the next section, as GIS plays a role.

7.2.6 Crop specific models

Many models have been written that are crop specific, and can model a crop's peculiarities. These were not given serious consideration in this project, other than to note their existence, as the ultimate aim is of course to model several different kinds of crop across the landscape. For examples of crop specific models and their use see Bellmann et al. (1986), Childs et al. (1977), Tscheschke and Gilley (1979) and Jones and Kiniry (1986).

7.3 GIS and crop models

The impact GIS can have on crop modelling can be broadly divided into two categories. Firstly, by providing an interface to the underlying models, which is a way of using existing models in a spatial context by supplying spatial varying data, and secondly by being an alternative paradigm to guide the development of models. It is fair to say that it is only in this first category that significant progress has been made, this is an example of the phenomenon that Wilson (1996) noted (i.e. that GIS had not as yet had a great impact on changing the underlying paradigms upon which environmental process models are based, as discussed in Section 3.4.1). In the case of crop models and GIS, the reason for the lack of impact can be explained by the ease with which existing crop models can be combined with GIS.

It would be fair to say the vast majority of crop models are point models, and there is assumed to be no interaction with adjacent crops and conditions at adjacent points — i.e. processes in adjacent areas do not affect the input parameters of the site in question. In order to link a model with the GIS it is simply necessary to run it repeatedly at each point across the area of interest. In a raster GIS this could be done in each pixel, in a vector GIS in each polygon. Whether these operations are performed sequentially or in parallel is a detail of the implementation and will not affect the model output.

There are many examples of the first category of linking in the literature. Maas and Doraiswamy (1996) used a very simple model, similar to Hanks and Hill (1980) see Section 7.2.2, but one that is set within a raster GIS. This means that the model can be applied to spatially varying

input parameters. They were also able to use remote sensed images for calibration. The output was spatially varying estimates of actual evaporation and biomass.

The most obvious use for GIS is estimation of input parameters for the model. If the model is to be applied at a site without a meteorological station it may be necessary to interpolate temperature, precipitation etc., from several remote sites. For example Collins and Bolstad (1996) explored and compared various ways of interpolating temperature data, using such methods as kriging and cokriging (see Section 5.7 for a discussion on kriging and cokriging). On a continental scale Kesteven and Hutchinson (1996) interpolated maximum and minimum temperature and precipitation — essential to this interpolation was a DEM which had a significant affect on all these variables.

Schädlich and Mauser (1996) linked an evapotranspiration model PROMET (based on Penman-Monteith, Equation 7.7) to a raster GIS. They generated spatially varying images of actual evapotranspiration over an area of some 100 km².

Examples of the second category of linking are less easy to find but a few can be identified or suggested. A classic operation in GIS is locating the ideal site for some facility. The usual example used is that of a landfill site or a leisure facility. In order to find the ideal site GIS layers describing distance to nearest road, population density, land cover, slope, soil type and many other factors are brought together in a process of cartographic modelling (Section 3.3.3). There will only be a few locations where all the criteria for the site will be satisfied. Booth (1996) reported on the use of a similar process for locating where different crop species could be expected to grow and how well they would grow. Initially the method relies on maps of climatic variables. The process begins with climatic maps of annual precipitation, maximum and minimum temperature etc, which can be used to select tree species appropriate for a particular region. Taking this a step further, Booth modified the model PlantGro (Hackett, 1991) so that it could be used over many grid squares to predict how well the selected tree species would grow — but this latter stage is again an example of the first category.

The concept of linking crop models and GIS is still developing. To date all the examples found in the literature have essentially applied the model independently at each point in the landscape, i.e. category one (e.g. Schädlich and Mauser, 1996; Booth, 1996); input parameters are spatially varying, but at any point the model is conceptually the same as the examples discussed above (Section 7.2).

However, with GIS there is an opportunity to model processes that have a spatial component, though how this could be used to change the paradigm of crop models and produce a practical improvement of modelling capability is hard to imagine, excepting erosion modelling. One possible area is that of the estimation of evapotranspiration. The Penman and the Penman-Monteith equations given above assume that the evapotranspiration is being estimated at a point in a very large region of constant properties such as roughness, canopy moisture etc. The area is assumed to be large enough that the atmospheric deficit has reached a steady state as the wind passes over

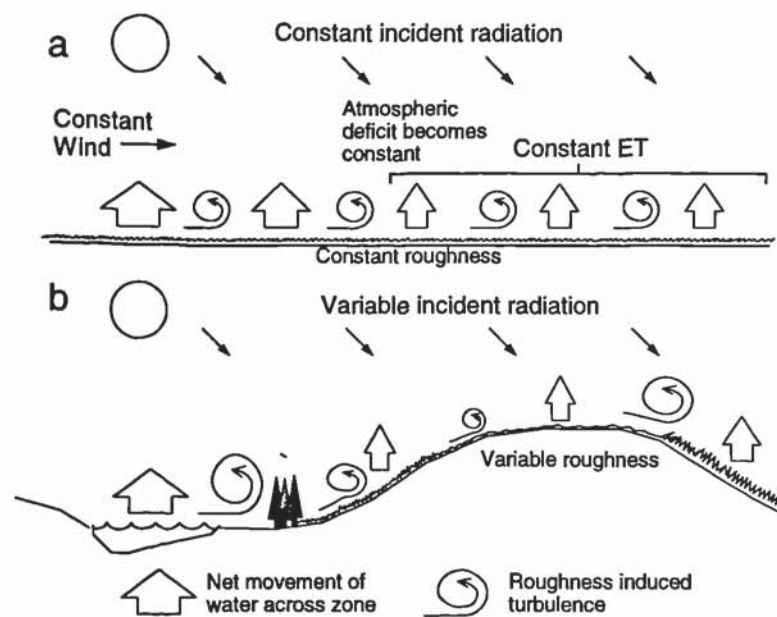


Figure 7.6: Alternative paradigms in the estimation of Potential and Actual Evapotranspiration. The atmospheric demand for water depends on how much that demand was satisfied upwind. In (a) ground conditions are assumed to be constant over a wide area so that the upwind evapotranspiration has had time and space enough to reach steady state conditions with downwind evapotranspiration. (b) In reality the landscape is complex and a constant atmospheric deficit is not reached.



Figure 7.7: Elements of remote sensing, after Elgy, Charnock and Hedges (1994).

the region Figure 7.6.a. In reality it is known that the landscape varies in roughness, and that the atmospheric deficit will vary over the landscape. Is it possible that in future spatial models of AE and PE will incorporate wind flow vectors, variations in roughness, crop type and soil moisture etc., across the landscape, Figure 7.6.b.

7.4 Remote sensing of the distribution of crops

Hedges (1989) required a map of the vegetation of the Tern region for his analysis. To obtain this map Hedges undertook a land based survey in the summer of 1980. A digital version of this survey is illustrated in Figure 7.10. An alternative approach would have been to use remote sensing. There are many good books and articles regarding remote sensing so the principles not be review extensively here — see for example Elgy, Charnock and Hedges (1994). Briefly, remote sensing, in this context, is the measuring of landscape properties by sensing emitted or reflected electromagnetic radiation at a remote point using a sensor mounted on an aeroplane or satellite (Figure 7.7).

Remotely sensed images are obtained from a wide variety of platforms, at different spectral and spatial resolutions. LandSat images have pixels of approximately $30 \times 30m$ and each scene is sensed in several bands: blue, green, red, near infra-red, infra-red and thermal infra-red. LandSat images were considered appropriate for this project, given that the groundwater modelling was

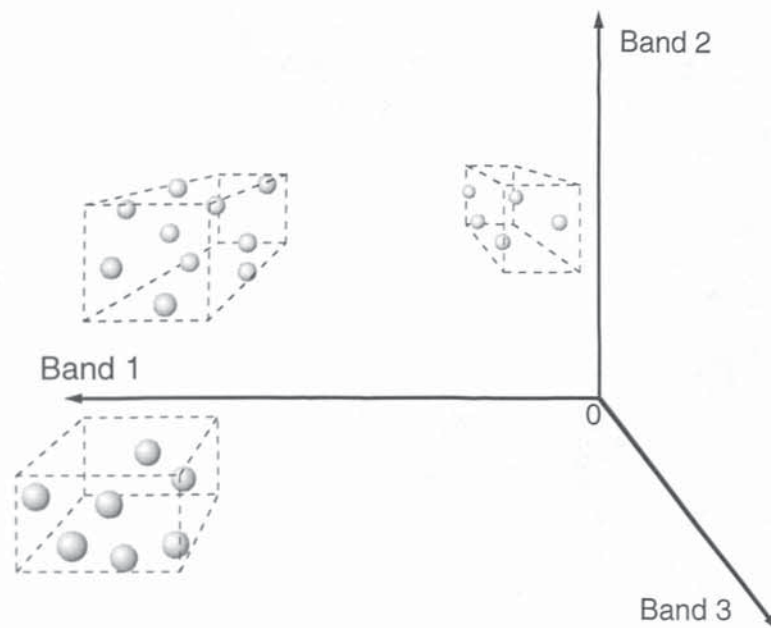


Figure 7.8: Pixels of the same land-cover will tend to cluster in the multi-dimensional space of the different remotely sensed bands.

performed on a $50 \times 50m$ basis, the DEM comes in a $50 \times 50m$ format, and the fields of the region tend to have dimensions larger than $30m$ square — the image is not therefore dominated by mixed pixels.

A Landsat image for 1990 was available and using ground truth statistics¹, several land cover maps were generated for the Tern region using the classifiers provided within the GRASS environment. A classifier is an algorithm that assigns pixels to class groups on the basis of its reflectance in the different wavelength bands, as detected by the sensor. In the simplest classifiers the bands are treated as dimensions, thus all the bands together form a multi-dimensional coordinate system. Assuming that a land cover type will have a typical “signature” of reflectances in the different bands, then pixels covering that land cover type will tend to cluster together within that multi-dimensional space and can be distinguished from other pixels, Figure 7.8.

There are two main classifiers available in GRASS, one is the classic maximum-likelihood (GRASS command *i.maxlik*) classifier. This assigns pixels to the most likely class on a pixel by pixel basis (Westervelt et al., 1993). The second is a Sequential Maximum Aposteriori classifier (SMAP, run with the GRASS command *i.smap*). This classifier improves classification by recognising that adjacent pixels are likely to be in the same class. It operates by analysing the image

¹ Ground truth statistics were generated in the Department Of Civil Engineering, Aston University for another project.



Figure 7.9: The SMAP classification. The black line shows the 1980 survey region (Hedges, 1989), in Figure 7.10.

at several spatial resolutions (Bouman and Shapiro, 1992) by dividing or segmenting it and not just analysing the individual pixels. The SMAP classifier has proved to be the best.

Figure 7.9 shows the SMAP classified image. For a crop modelling approach, where each crop has different physiological parameters that influence growth and response to moisture, the full range of crops provided by the classification can be used. For a cartographic modelling approach such as Hedges's (1989) technique (Chapter 4) where the principal crop parameter required is rooting depth then the crops can be grouped and reclassified into a lesser number of root depth classes.

Even the SMAP classification is not perfect, no classifier ever will be. To some extent this is due to a mismatch between what is wanted in a map and what can be got from what is essentially a photograph. A map has already been referred to as a model (Section 3.1), so a displacement of



Figure 7 10: Hedges's (1989) 1980 agricultural survey.

x on the map corresponds to a displacement of X in the world (ignoring the complications of map projections) As a model, a map has inherent simplifications, which are explicitly and implicitly decided upon by the cartographer. Simplification in a map is undertaken on a feature by feature basis: road features may be widened, straightened, missed out and moved; coastlines are put at mean high or low water line etc. An image or photograph also has simplifications, but these result from the capabilities of the sensing technology. In general the simplification occurs at a certain length dimension, below which features, no matter what they are, or their significance to the task in hand, tend to blur.

However, there is still room to improve the classification, and it has been postulated that other spatial data can be included in the classification. If the amount of light in the red band is a suitable dimension, why not the elevation of the pixel, or the distance from the river — both of which can be expected to be correlated with land use? In Figure 7.10 it is clear that permanent pasture tends to occur next to rivers. This may be because there are waterlogged conditions unsuitable to other crops, or possibly because pasture is less vulnerable to flooding, either way it is a relationship that could be exploited to improve the classification.

Hedges's (1989) survey, whilst not having great resolution of crop types (in fact only 9 categories are recognised) has the useful property that it covers a region completely. Commonly ground surveys pick survey points haphazardly across the landscape, often very far apart, and only a very few for each landscape type. Hedges's (1989) survey provides a useful dataset for testing how variables such as height, distance from river or slope might be introduced into the classification in order to improve it. When using a more widely distributed set of survey points not only does one need a widespread digital elevation model or river network, which would be expensive, but

also there is the possibility that some unforeseen bias is introduced by the selection of ground survey samples. For example ground survey sites might be expected to be close to roads, which themselves might tend to be near the bottom of valleys, i.e. close to rivers and at low elevations. Similarly, when selecting ground truth sites one is generally looking for an example of each ground cover type of interest. Often a few ground types dominate an area, but this domination may not be represented in the relative sizes of the training areas. A spatially continuous survey such as that of Hedges (1989) circumvents these problems.

The National Remote Sensing Centre (NRSC) were unable to provide an image of the Tern area for 1980 so a full classification was not possible. However, samples of elevation, distance from river, slope and landcover were extracted and some preliminary statistics were performed. The results illustrate that some of these variables might well improve a classification (see Table 7.1), particularly in distinguishing permanent pasture (as was expected from the visual inspection of Figure 7.10). As the survey was conducted over a small area and consequently includes little variation in height compared with what might be expected across a full LandSat scene, we can expect the effect to be greater in a full scene.

7.5 Implementation of a crop model in GRASS

The crop model was the last part of the environmental system examined, consequently not much progress was made. Crop models tend to be simpler than soil moisture or groundwater models, and where code is provided, this is usually because the model overlaps with the soil moisture component to a large extent. To demonstrate that a link could be achieved with the ease claimed in Section 7.3, van Keulen's (1986) model (Section 7.2.5) was implemented as a shell script largely comprising calls to *r.mapcalc* (the code for this is documented in Appendix A.2). It has been tested on the example given by van Keulen but has yet to be used on the Tern area.

This is an example of the first category of linkage discussed in Section 7.3. The only problems encountered while coding are that *r.mapcalc* does not have a built in "look-up" function and, as described in Section 7.2.5, the model makes use of a considerable number of data tables in order to keep the model simple and allow it to be performed on a calculator. Three look-up functions were added to *r.mapcalc* as described in Section A.3.

The link here is similar to the Modflow link and the SWMS_2D link in that rasters are used to hold the input and output data. The input and output raster names are pre-set and the onus is upon the user to make sure rasters of the correct names exist within the database and contain the correct information. This means that the user does not have the flexibility to logically link any raster with a model parameter — this could be implemented in future prototype.

It is the simplicity of the crop model that enables it to be implemented largely within the bounds of such a restricted language as the *r.mapcalc* syntax. The advantages of doing this are that the programmer is saved from all the tedious administrative tasks (file manipulation, layers

Land cover	Elevation	
	mean	stdev
Cereal	71.77	9.87
Sugarbeet	75.78	17.62
Potatoes	76.75	6.20
Permanent pasture	69.16	11.65
Ley pasture	74.81	7.68
Woods	74.48	12.80

Land cover	Distance	
	mean	stdev
Cereal	1208.87	882.71
Sugarbeet	1498.57	998.77
Potatoes	1528.57	810.81
Permanent pasture	854.60	845.59
Ley pasture	1441.96	966.82
Woods	1716.79	685.02

Land cover	Slope	
	mean	stdev
Cereal	25.67	13.10
Sugarbeet	22.03	12.05
Potatoes	32.76	13.00
Permanent pasture	16.45	15.57
Ley pasture	28.29	15.53
Woods	23.82	11.94

Table 7.1: Statistics of elevation, distance and slope grouped by land cover. Note that in all cases there is a clear difference between the position of the mean of permanent pasture and the means of the other variables. Also note that the standard deviations indicate that there is overlap, thus none of the variables could be used in its own to distinguish permanent pasture but they could be used within a classifier along with reflectance bands to improve the classification.

of error checking etc.) that would be implied using a more general language.

Ultimately it would be desirable to replace the simplistic soil moisture model used by van Keulen's (1986) model with the SWMS.2D model. There is no practical reason why this can not be done. It has already been stated in Section 3.5.1 that a controlling process is needed in order to bring different models together within a single system. Shell script languages were identified as appropriate tools with which to construct such processes. Thus the script which implements van Keulen's (1986) model could be modified, with those sections concerned with soil moisture and the position of the watertable replaced with calls to the *r.swms.region* and *mod.write* and *mod.run* programs (Sections 6.5.2, and 5.4.3).

7.6 Evaluation

At this stage of the project the aims were to explore the issues surrounding the use of crop models with GIS. Many of the issues are the same as those encountered with groundwater and soil moisture models discussed in the preceding two chapters. Those issues that are the same will not be reiterated and those areas where crop models differ from will be discussed and emphasised. The most obvious difference is that the model includes a biological entity — a plant. As the least progress has been made in this aspect of the project, the conclusions presented here are tentative, however they are useful and add another dimension to the concluding discussion of Chapter 8.

As with other disciplines that use environmental models, there has been some progress in the discipline of plant science to link GIS and crop models and, like these other disciplines, the impact of GIS on crop models has been largely in the data handling and parameter generation, but rarely on the basic paradigm guiding the development of the models.

7.6.1 Plants as *model-able* entities

Plants are biological entities and consequently considerably more complex than groundwater system or even soil moisture. They modify their behaviour to enhance their survival chances. Even crops, which over generations have been selected for other characteristics — yield, synchronous growth stages, resistance to pesticides etc. — still have scope to modify their behaviour. Loosely speaking plants are able to “cut their losses” when conditions depart from the conditions that they have been selected for. Of course when most crops encounter a moisture deficit the stomata begin to close, but that is a short term response, how plants behave when they encounter chronic conditions is less well understood. So when a continuing deficit is encountered we can expect a crop to respond in different ways; by preferentially taking water from the more moist layers, perhaps by allocating more of the assimilates to root crop to penetrate deeper soil, or by some other behaviour. However, it is clear that exactly how, what and when different crops react is still a subject of research and crop models only take into account these postulated effects in very basic ways.

Crops are often treated as the same basic crop, but with slightly different parameters for common attributes (though, as noted in Section 7.2.6, there are many crop specific models available). Hanks and Hill (1980), in their model, assume that a crop will transpire at the maximum rate if any of the soil layers is above the critical threshold of moisture content, regardless of how dry other layers are, hence, after a brief rain fall transpiration is at a maximum fed from the very top soil layer. In van Keulen's (1986) model, root growth is allowed for, but this is assumed to proceed at a crop specific constant rate. It would not require much ingenuity to allow the rate of root growth and the partition of assimilates to change in response to moisture conditions. The author attended a seminar² on this subject and discovered that the answers to many of these questions, like the method of signal propagation, are still very much at a research stage.

7.6.2 Conclusions

As with many models in general, it is often a case that the crop model complexity outstrips the available data. This is a theme that will be returned to in Section 8.5.1. Furthermore many of the problems are similar to those discussed with regard to soils, as there is considerable overlap. When it comes to getting the spatial distribution of crops over a region, unless some form of central management or registry is in place to provide the data, then remote sensing represents the most cost effective and quick way of acquiring this information.

Though only an initial link to a crop model was performed, this demonstrates the ease with which such a task can be achieved. The simplicity of the model meant that the effort in writing the model from scratch was equivalent to linking in existing model code. Clearly considerable work still needs to be done on this aspect of the project, but some of the major problems have been highlighted by this and the previous chapter, and these will be discussed further in the next chapter.

² Discussion meeting, "Water Uptake by Vegetation", British Soil Water Physics Group, Department of Soil Science, Reading University.

Chapter 8

Discussion

8.1 Introduction

It became apparent as the project progressed, that the original aim of fully implementing a complete modelling system and simulating the Tern groundwater, soil moisture and crop system satisfactorily was too ambitious to be achieved within one three year research project (e.g. the project described by Ireland (1995), Section 3.4.2, required 75 man years). The result has been a project that is rather broader than it is deep, a great many subjects have been touched on, and it has been necessary to leave some clearly useful and interesting avenues of research only partially explored.

This project is based on computers. This in itself presents a problem given the speed with which computer technology changes. During the course of this project many changes have occurred. The World Wide Web has achieved dominance as one of the most used and apparent of the Internet devices. Microsoft introduced Windows 95 and Windows NT and even in the last few months the language Java is reported to be challenging C++ as the main systems programming language. The list of changes; incremental, exponential and revolutionary could go on, suffice it to say that the tools chosen for this task in 1993 would probably not be the ones chosen for the task now.

Nevertheless the results of this project are still useful. Any organisation or individual, faced with an ever changing vista of computing potential; faster machines, cleverer operating systems and speedier communications and more versatile software, must at some point decide to realise that potential as a set of hardware and software that will persist unchanged for some significant amount of time. One cannot always wait for the next development. Lessons will be learnt from that realisation and then, after some time, a new higher potential may be realised and the older technology is thrown out. The rate of innovation caused by so many people, so much time and so much money being expend on all the facets of the computer industry can make the change seem continuous, but for any organisation or individual change proceeds as steps, often as very large ones.

At the starting point in 1993, an environmental system made up of several subsystems needed to be modelled, a level of technology was available and this project began.

The research and exercises have been described in the preceding chapters. This chapter sums up the work of the preceding ones, explains what has been learnt and how this might be applied in future projects. Firstly the overall strategy of exploiting existing models, linked via a GIS and using secondary data, will be evaluated. Secondly a conceptual design of the next prototype based on the lessons of this project will be given. Thirdly the lessons learnt from this project will be placed in an organisational context. Fourthly the problems of data will be discussed, and fifthly an overall evaluation will be drawn.

8.2 Evaluation of the overall strategy

From the start money and time have been limited, so the strategy, laid out in Section 3.5.1 has been to use existing components as much as possible and to link them together via a GIS. Little effort has been expended on the interface or on help facilities. Similarly little expense could be devoted to obtaining high quality primary data, so the strategy included the use of secondary and surrogate data. Has this been a successful strategy?

Even with this strategy it has not been possible to complete the project as originally envisaged. But some useful discoveries have been made. The models chosen have probably been over complex, not in relation to the problem, but the time available. This has meant that after they have been linked with the GIS there is not enough time to go through the process of parameterisation let alone using them together in an integrated modelling effort. However, even less would have been achieved if models of this complexity had been coded from scratch internally in the GIS.

Part of the strategy has been to use a GIS to provide the heart of the data handling tasks the project has required. The GIS proved well suited to the task of performing Hedges's (1989) original analysis technique, Section 4. Assembling the data and getting it into the right form for the models has been a major part of the project, and there is little doubt that GIS helped in this. For example by combining vector maps of river networks and fault patterns in order to generate the raster describing the groundwater model boundaries, and there are countless cases when GRASS commands were used to capture and manipulate data:

- the *v.digit* program for digitising paper maps;
- the *r.mapcalc* command for manipulate the data, e.g. changing units etc;
- several importing routines; *r.in.ascii*, *r.in.sunrast*, *v.in.ascii*, *v.in.dxf* and *v.in.arc*;
- the visualisation commands; *d.rast*, *d.vect*, *d.sites*, *d.histogram* and *d.3d* etc;
- image processing facilities; *i.mazlik* and *i.smap*;

- several interpolating routines; *s.surf.idw* and *s.surf.tps* for interpolating site samples to surfaces and also *r.surf.contour* which generates a surface from digitise contours;
- the watershed analysis program (*r.watershed*) proved very useful for generating hydrological features, basins, river networks etc. from the DEM.

To attempt this without having the existing GIS functionality to hand would have involved considerable effort and time writing a great number of *ad hoc* programs and scripts, and moving data between several different packages. However, there were many cases where the functionality fell short, the obvious are:

- statistical and geostatistical analysis had to be performed by external packages;
- some data formats were not supported e.g. Ordnance Survey's National Transfer Format (NTF) (OS, 1993) — not surprising in American software;
- database query facilities — the package *RDB* had to be used to manipulate the large watertable readings dataset.

Where functions were missing it was necessary to write programs and scripts to get around the problem. Generally it was easier to use existing more primitive GRASS programs to do this or to write functions to export the data to some other package, rather than write a program to perform the function from scratch. Thus, when data importing functions were required it was easier to write scripts using *awk* or *PERL* to convert the files to a form so that the ASCII import functions could be used (*r.in.ascii*, *v.in.ascii* and *s.in.ascii*), rather than writing a program to do the whole task. Similarly no attempt was made to write statistic and geostatistical functions, simple functions were written to import and export the data to other packages. Much of the shortfall was due to the purposes for which GRASS was created in the first place (Section 3.5.2), a different GIS package would have a different set of functions, though in general there is a core set of functions which are common to all GIS. Thus a GIS provides a useful set of general purpose functions with which to begin to work on a project and should also provide the facilities to augment this set for a specific purpose.

The final part of the strategy was to use secondary and surrogate data, this is a huge subject in itself. This project has encompassed the examination of statistics and geostatistics for estimating watertable distribution (Section 5), the use of regression equations and soil maps and surveys to provide parameters for soil models (Section 6) and remote sensing for crop distribution. In practical terms one decision proves to be difficult, is deciding where to stop and when to state "this is the method that will be used" in order to generate or improve the data for a project. For example in predicting watertable distribution many different relationships between various explanatory variables can be discovered, and many different models could have been fitted to geostatistical semivariograms and cross-semivariograms, but which is the correct and which should

be adopted to generate a consistent set of surfaces. In this project a procedure using *s.surf.idw* was adopted as a standard for generating consistent watertable surfaces.

8.3 System design of the next prototype

As stated in Section 3.5.1 a prototyping approach was adopted for system development. The principle behind the prototyping approach when applied to information systems, is that a working system is produced that the user can experiment with and comment on and so provide the basis of a design for the next prototype. Thus prototyping is an iterative process.

Bearing this in mind this section is intended to define the next prototype in the light of lessons learnt from experiences to date.

The link between GRASS and Modflow was the first to be constructed, and represents not only a first prototype in a process of development, but also a process of learning — learning and improving general programming skills, and learning the data structures, programming libraries, conventions, capabilities and limitations of GRASS. In addition the link was first programmed into a version of GRASS that did not have floating point facilities, thus complications were introduced that subsequently proved unnecessary (particularly the introduction of multipliers and divisors). Because of the lessons learnt with the Modflow link, the link between SWMS_2D and GRASS was constructed more quickly and in a different way.

8.3.1 User interface

When the Modflow link was first designed it was envisaged that the specification of the model parameters and geometry would be an interactive process. GRASS has a library of functions for constructing a form based approach to the user interface, the so-called VASK library. More sophisticated tools exist for constructing interfaces (one such language, Tcl-tk, was briefly experimented with in Section 6.5.3), but it was decided to use VASK as it was very simple (appropriate for a prototype) and fully integrated with GRASS (there are some examples of VASK forms constructed in Appendix B).

VASK proved reasonably able to cope with editing the raster lists, the heart of the link between GRASS and Modflow. Where it proved less appropriate was in the specification of the model geometry, particularly in the specification by the user of the irregular grid. In this case a more graphical interface would be appropriate. VASK has not ported with the new commercial version of GRASS, Grasslands (see Section 3.5.2), presumably it is not considered sufficiently attractive or modern for a commercial package.

It was later realised that there were occasions when it would be useful to be able to run parts of the Modflow link as a command line, and thus enable it to be embedded within higher level commands (see for example the calibration script in Section 5.8.2). The interactive form filling approach of VASK was not suitable for this, but the GRASS parser proved more useful.

The SWMS.2D link did not use VASK, instead, realising that the link would be embedded within at least one higher layer of programming, the GRASS parser was used as the interface from the start, (see Shapiro et al., 1993). The GRASS parser allows the user to either access the program by specifying command line arguments, or, if just the program name is given, by answering a series of questions. In the SWMS.2D link an interactive way of editing the dataset was also provided, one which simply used a text-editor as an user interface — this proved simpler than designing a form interface.

It was a design decision at the outset not to attempt to build knowledge or expertise into the link. However VASK and the GRASS parser do allow defaults to be built into the required input parameters; good defaults values are very useful to the user, and can be considered an elementary form of help.

Returning to the discussion of Section 3.4.3 where a distinction was drawn between interface and linkage construction tools that are supported by the GIS and those that are supported by the underlying operating system (OS). It should be clear that the interface constructed was actually a mixture of the two — VASK and the parser are GIS tools, the various programming and scripting languages are OS tools. The mix is most apparent when you consider the C programming language which is the basic system programming language of UNIX but GRASS extends its capabilities by providing libraries of functions. This mixed approach has proved flexible and useful as a learning exercise — though if the system were to be developed into a finished product it would be preferable to use a less diverse set of tools to make maintenance and development more convenient.

8.3.2 Reconciling differing data models

In Section 3.3.1 the data models upon which GIS are generally based were described and it was also noted that computer implementations of environmental process models also have data-models. The representations of reality used by GIS and process models are developed in response to the priorities of the GIS and the model designers, and can be very different, see Figure 8.1. Linking a process model to a GIS means reconciling these different representations as expressed as a data model. The current generation of GIS have 2 Dimensional (2D) functionality with time and depth relegated to simple attributes. Generally process models use the time dimension but have any combination of spatial dimensions. Hazelton (1991) suggests two ways of dealing with this incompatibility. Firstly to use “ad hoc” solutions and “complex linkages”, and secondly “to develop a 4D GIS” which could interact directly with the models. Clearly the approach adopted in this project has been of the first type. With a few exceptions, 3 and 4D GIS are still at a research stage (so-called Geo-Scientific Information Systems, GSIS, were briefly discussed in Section 5.3.1), and 2D GIS dominate in operational environments. Given the level of investment and the predominance of 2D GIS applications, this is likely to remain the case for a long time. However, environmental problems must be addressed now, which means that, despite the drawbacks, 2D GIS and process models must be linked.

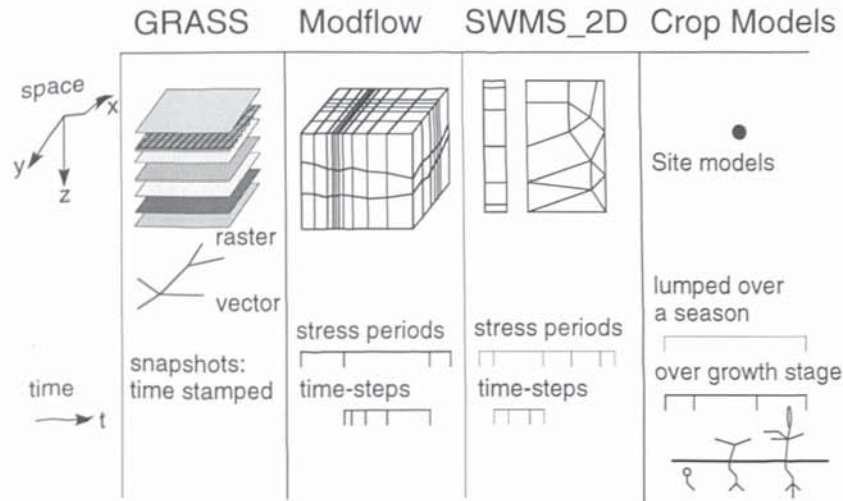


Figure 8.1: Different representation of space and time in GIS and Models.

Reconciling the 2-dimensional spatial models has proved quite straight forward. Modflow and GRASS use very similar models for 2-dimensional space. Some resampling functions were required to move between the irregular grid of Modflow and the GRASS raster, and rasters had to be linked together to represent the vertical dimension. SWMS.2D was linked at two levels; a low level basic link was developed so that the model could simulate conditions at one point on the landscape. The basic link was written so that it could be embedded within a higher level process that controls the way in which it is applied to locations across the landscape (Section 6.5). The crop models looked at would be linked in a similar way to SWMS.2D. Generally crop models have much simpler models of space than groundwater or unsaturated flow models. It was thus possible to implement one model simply as an *r.mapcalc* script (Section 7.5).

Handling the vertical dimension has proved more problematic, see Figure 8.2. Within the current generation of GIS, GRASS is not exceptional in not handling fully 3-dimensional data. The current generation have been described as handling $2\frac{1}{2}$ dimensions (Figure 8.2.a), i.e. they can cope with simple elevation data. However, these GIS are not able to properly handle fully three dimensional cases where, for example, the landscape forms an overhang (Figure 8.2.b) and they are certainly not able to handle complex geological or mining situations or building plans with more than one storey (Figure 8.2.c). A solution to this problem is to build up the 3 dimensional representation as layers, which was the solution adopt here. In more complex geology the layer representation would be less amenable, whilst for soils this is not a problem.

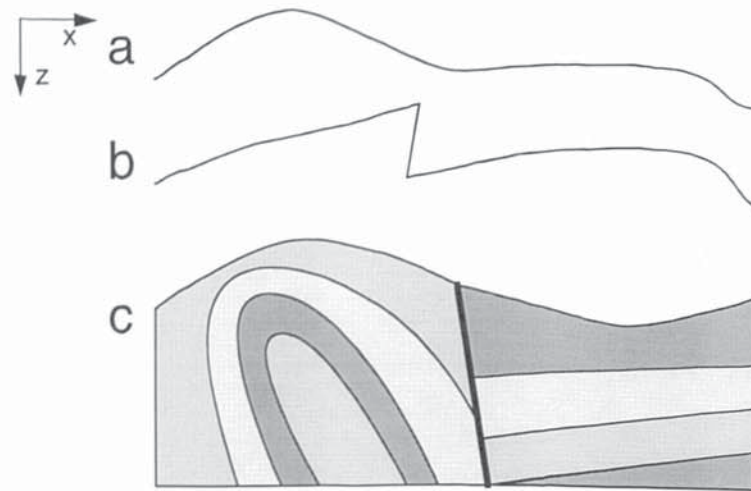


Figure 8.2: Handling the vertical dimension in GIS, (a) $2\frac{1}{2}$ dimensional representation of elevation, (b) an overhang situation that cannot be represented with $2\frac{1}{2}$ dimensions, and (c) complex geological situation, simple sedimentary sequence on the right can be represented easily using layers, the anticline cannot be easily represented.

Environmental process models usually include a model of time. Generally this comprises periods across which time varying boundaries are assumed to be constant. Periods may be divided into steps and sometimes the numeric solution involves further subdivisions. However, the current generation of GIS, GRASS included, do not handle time in a very sophisticated manner — if at all. Generally rasters or vectors, when they describe some time varying variable, are considered as snapshots of that variable at a point in time. Several authors have considered the issues of temporal GIS or spatio-temporal information systems, see for example Kowalczyk and Kemp (1993). Given that such a system is not available then the most common way of handling the time dimension is to place database objects, whether raster or vectors, into a sequence.

This was the solution adopted in this project for both the Modflow and the SWMS_2D link; rasters were linked together within the dataset object to form a time-series, Figure 8.3.a, and this proved adequate for the task. The only major drawback encountered was when the number or length of the periods of a model were changed for some reason, when this occurred then the rasters had to be laboriously reassigned to the correct periods.

An alternative approach would be to link the rasters together to form a time-series independently of the dataset specification (i.e. as a database object in its own right, Figure 8.3.b). And then link the dataset to the time-series. As a database object a time-series would need a family of functions to handle it. For example, to display a point in time, one would need to specify, not only the region over which it was to be displayed, but also a point or period in time, if no

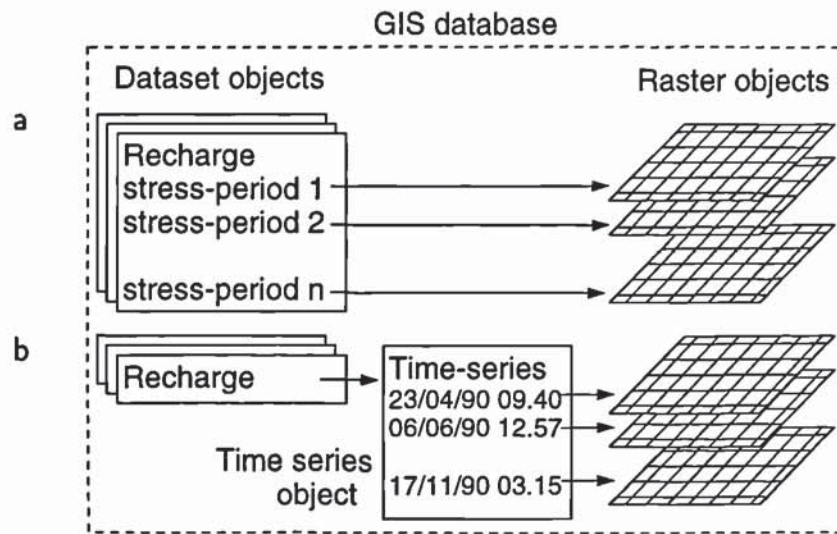


Figure 8.3: Approaches to handling the temporal dimension.

raster existed in the time-series that represented that particular time period, then one could be interpolated from those rasters that are nearest in time, analogous to the way rasters are spatially resampled when the resolution of the region does not match the raster's internal resolution.

If such a scheme was implemented at a GIS level, i.e. it was a dataset object available for any function to access and use, it would be a very useful and powerful facility. For example one can imagine an *r.mapcalc* command which not only used an offset in space, i.e.

```
r.mapcalc "surface_smoothed=(surface+surface[0,1]+ \
surface[1,0]+surface[0,-1]+surface[-1,0]) / 5"
```

but also allowed offsets and absolute positions in time, e.g.:

```
r.mapcalc "time_smoothed= (time + time[1] + time[-1]) / 3"
```

where *time* is the name of a time-series made up of sequence of rasters, in this hypothetical set of functions *time[1]* generates a raster from the time sequence representing conditions at a time 1 unit, before or after the current time (i.e. the current active GRASS time in the same way GRASS has a current active region.)

In terms of database management there would be advantages to an arrangement such as this. data compression could be improved and the problem of handling numerous rasters is diminished if they can be linked together. GRASS uses run length encoding to compress rasters, this exploits the property many variables have that the values at one point are often the same as values close

by. This property often holds through time as well. To implement this model of time would require considerable extensions to the internal GRASS structures. Such a development is the responsibility of the GIS developers, it was not the purpose of this project to rewrite the internal structures of the GIS.

To return to the problem of handling time in the link between GIS and models. Even if such a model is not implemented within the total GIS framework it is still worth separating the user defined periods of a model from the time-series of input parameters and to implement the temporal resampling described above within the model link.

Actually there is already an object within the GRASS database which was implemented after the underlying data models were decided upon, that exhibits some of the properties described above. This object is the so called *group* structure, which is used by the imagery programs (the *i.* programs Westervelt et al., 1993) and links rasters that might represent different bands into a *scene*. Other information such as training area, land use class signatures and rectification points are also stored in the group object. Because the rasters are just normal GRASS rasters (all the other information is stored in the group object) any raster can be included in a group — this has been used to advantage to rectify scanned maps to their correct geographic co-ordinates.

A similar scheme to that outlined above for the temporal dimension could also be implemented for the vertical dimension. However, when considering this possibility a difference between spatial and temporal data is highlighted, or rather a difference in the way variables are commonly measured in space and time.

The vertical dimension has been treated as a sequence of layers which, at first glance, resemble the sequence of rasters describing the variations in time. However, the vertical dimension is divided up into intervals whose size varies not only along the vertical dimension but can also vary with horizontal position. This is because the position of these vertical divisions is chosen to correspond to real physical boundaries — in geology the width of these vertical intervals is governed by the boundaries of geological units (Figure 8.4.a), in soil by the horizons — and thus the width of the intervals varies with horizontal position (Figure 8.4.b).

Temporal data is seldom sampled to correspond with a temporal boundary that varies in space, one could, perhaps, imagine a set of rasters which, instead of recording the amount of rainfall in each pixel at a given time, recorded the time at which a given depth of rainfall was reached. Similar depth data could be recorded in a way which resembled the way temporal data is usually recorded, by representing the geology or soil with horizontal planar *slices*, that describe the variations at a fixed depth (Figure 8.4.c). This would be a very tricky way of handling such data though it would have a use when the geology becomes too complex to be described using the layer approach (as for the anticline in Figure 8.2.c).

Largely the way data is measured and thus the way it is treated in information systems is due to the practicalities of measurement. In geological investigations boundaries are sought, these may be inferred from indirect evidence or directly observed with geophysical methods.

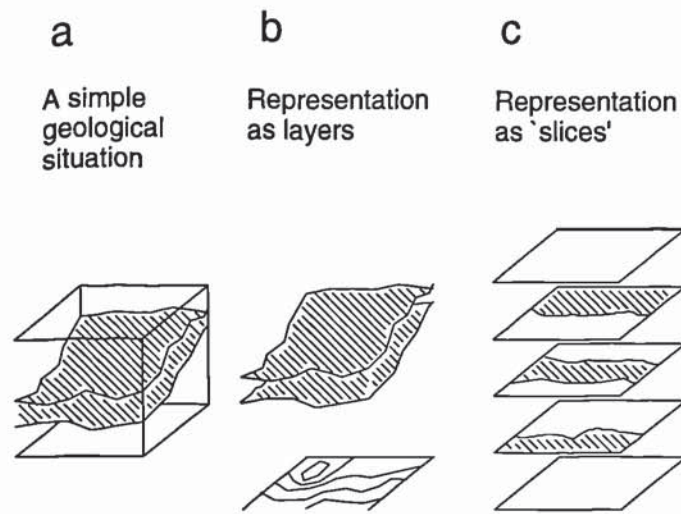


Figure 8.4: The vertical dimension, (a) a simple geological situation, representation by (b) layers – rasters are divided between those that describe the variation of depth of the layers and those that describe the variation of attributes across the layers and (c) slices: — rasters describe attributes as they vary at a fixed depth.

The above discussion has been about handling 4-dimensional data in what is essentially a 2-dimensional environment. This is a problem that is going to be an issue for sometime given the level of investment that has occurred in such systems. Evolutionary change in 2-dimensional GIS has occurred, particularly for handling the temporal dimension (i.e. GRASS has implemented a scheme of time stamping objects in the database), but these are generally clumsy add-ons that do not give the functionality available for the two horizontal dimensions.

The research into fully temporal GIS is still largely at a theoretical stage; see for example Kowalczyk and Kemp (1993) or Wachowicz and Healey (1993). Research has generally been aimed at solving the traditional database problems of: (i) allowing fast access but using minimal storage space, maintaining concurrency, preventing data duplication; and (ii) of developing a simple but rich and flexible query language. Problems which are of great concern when the GIS is used for asset management, such as might be performed by the utilities, by county councils or other large businesses. Only a few papers have touched on the issues of linking any kind of environmental model to temporal GIS, Hazelton (1991) is an example.

8.3.3 The next prototype for linking a process model and GIS

The above discussion shows that the dataset approach as illustrated in Figure 3.9 has drawbacks, Figure 8.5 shows a modified form of Figure 3.9, a different strategy for linking a process model

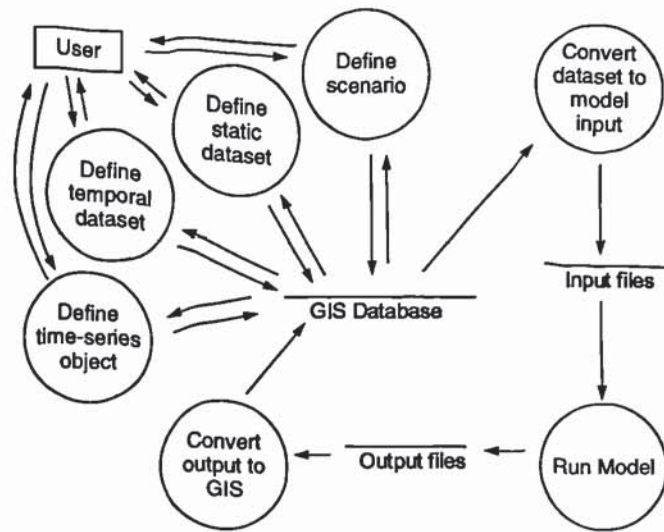


Figure 8.5: Modified system flow diagram using the same notation as Figure 3.9, the specification of the dataset has been split into three separate operations, also time-series are specified separately from the dataset specification.

with a GIS. In the approach illustrated in Figure 8.5, which takes into account the lessons discussed above (Section 8.3.2), the dataset is separated into separate categories of components;

- components that describe the fixed elements of the system; taking the Shropshire Ground-water Scheme as an example, these elements would be the properties of the boundaries and geological units, soil types, the location of rivers etc., i.e. parameters that can be expected to remain constant through time;
- those that describe the background time variant properties of the system (i.e. for the Ground-water system; time-series of precipitation, evapotranspiration, watertable position and river heads etc.);
- and those concerned with a particular scenario (e.g. pumping rates, river flow, crop types, irrigation rates etc. and also grid sizes, start and finish times, periods etc.).

The split between elements that could be described as part of a scenario to be tested, and those that are part of the background time variant conditions, is of course subjective.

Constructing a link to a model in this way would involve splitting the process of amassing the input data, but this too has advantages. Firstly, the fixed part needs only be specified once for a particular region, which could be done by an expert. The time variant part of the dataset could, similarly be constructed just once, or in an organisation if the expense was justified it could

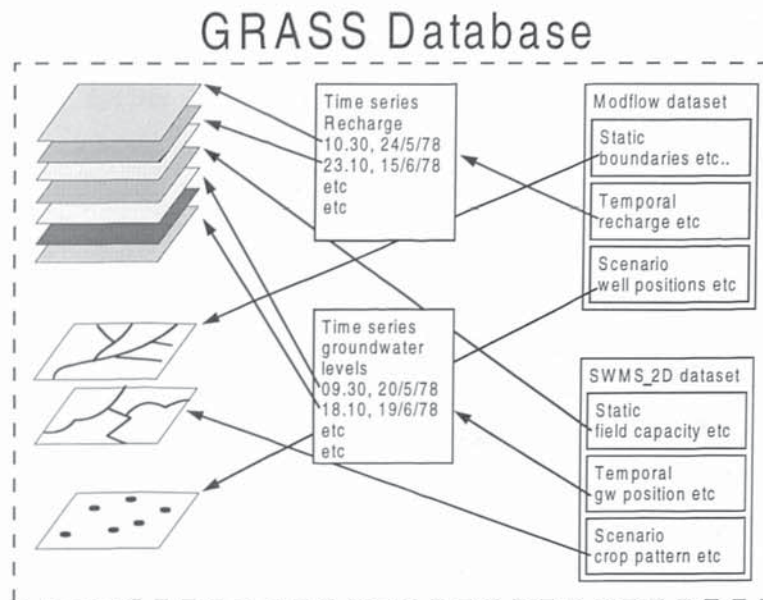


Figure 8.6: The way objects in the dataset would be logically linked together under the second prototype. Only a few example objects have been shown to keep the diagram clear.

be linked directly to the sensors monitoring the temporal variables of interest, in the case of the groundwater scheme these would be evapotranspiration, rain, watertable etc. That would leave just scenario components to be defined by the user which would depend on what was required from the modelling exercise. Figure 8.6 shows how the different items in the dataset would be linked together under this approach.

So far this description of a second prototype has just expanded the strategy for linking single process models to the GIS described in Section 3.5.1 and illustrated in Figure 3.9. A major concern of this project has been, of course, linking several models together to simulate the whole environmental system of interest, the original strategy was illustrated in Figure 3.10 and required a controlling process to call the different models at the appropriate times. This approach has proved valid as far as it has been tested and so remains unchanged.

8.4 A modelling system in an organisation

One can differentiate between two cases when using a modelling system to solve a problem: those where the project is an *one-off*, as this project has been, and those where the project will be ongoing. The strategy followed for this project clearly has advantages for the one-off project, but

does it have a place in an ongoing project — the author would argue that it does.

Attempting to design a whole system from scratch, delivering the system, and expecting it to answer all the users needs, is unrealistic. The fact is that the users rarely know exactly what they want from the system, and it is only when they have the system in front of them that they can see possibilities. This is the justification for many developers to choose the prototyping approach. In the longer term not only are the immediate needs of the users sometimes undefined, but the needs of the user five or ten years later are even more hazy.

The system undergoes a life-cycle, and whilst the uncertainty of the future means strict planning is not possible, it is possible to plan for change. A possible life-cycle plan if the loosely linked approach is adopted might be as follows:

1. The system begins just as an *off the shelf* GIS using the available data.
2. Tasks to which the system is to be applied are identified and where the GIS is lacking, other components are loosely linked into the system. If absolutely necessary customised functions are quickly and *inefficiently* coded into the system.
3. As the system is used, bottlenecks and shortcomings will become apparent. So, for example, in the system developed in this project, high-level PERL functions (such as the ASCII raster functions, see Appendix A.4) have been developed. These were quickly constructed but are slow. If these functions are used a lot then it may be worth re-implementing these as a PERL add-on module.
4. The tasks for which the system are to be applied are redefined, old models and functions are discarded and new models and functions are loosely added.
5. Again bottlenecks and shortcomings are identified and dealt with. The inner core of functions and links gradually expands.
6. After some time (say five or ten years) the internal structures and architecture of the system will be too much at variance with the present and predicted future requirements of the organisation. At which point the system is abandoned and a new GIS architecture is adopted.

This is illustrated in Figure 8.7.

An organisations needs are not static but will evolve with time. To maximise the benefits from the corporate data resources, from the GIS and from the use of modelling functions a policy of loose-linkage between the GIS and models (or other external applications) can be adopted.

8.5 Data

A great many issues concerning data have been explored during the course of this project; parameterising models, error, accuracy and various forms of secondary and surrogate data. It would be

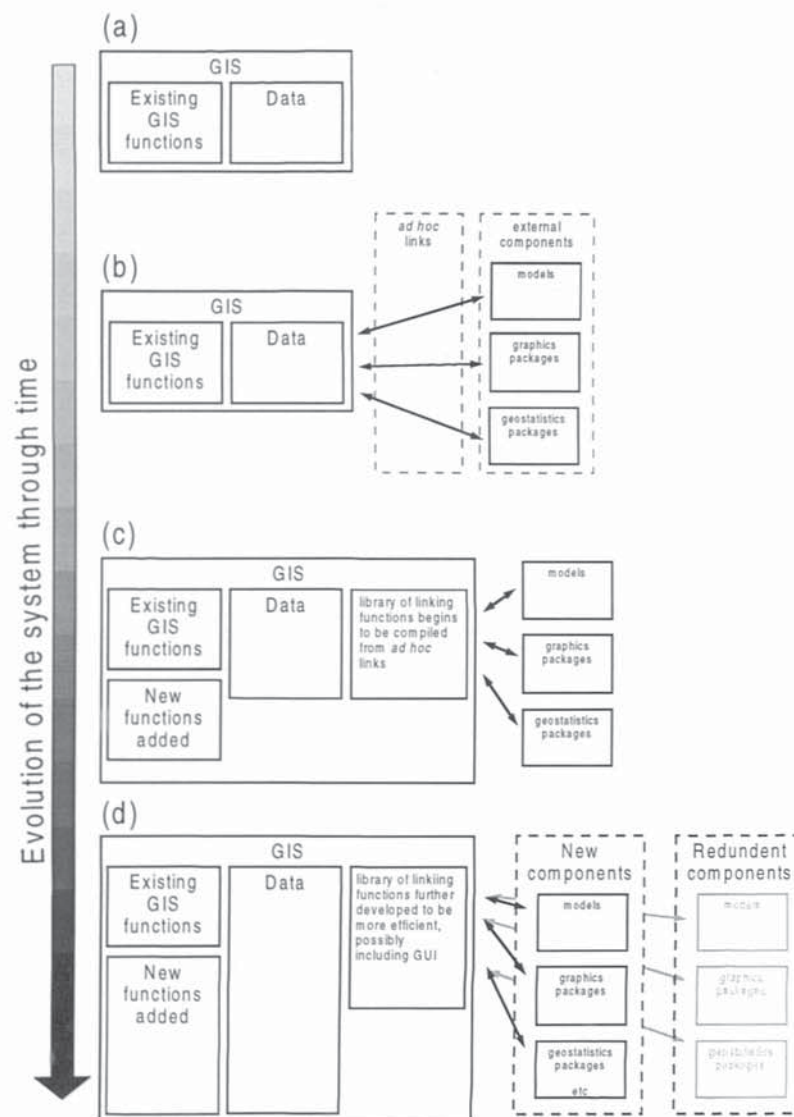


Figure 8.7: Life cycle of a loosely linked modelling system.

fair to say these the issues and problems have been the most difficult and time consuming part of the project.

8.5.1 Parameterisation

Parameterisation is a very useful term for an important activity. Obtaining good parameters which have physical meaning and cause the model to behave in a way that emulates the system of interest can be a very difficult, sometimes impossible, task. In some cases using the real measurable parameters from the environment do not enable the model to behave like the environmental system hence one is looking for an *effective parameter*. For example in Section 5.5.4 and Figure 5.25 the problems of defining an effective transmissivity are illustrated

Many authors have looked to GIS to provide this facility (many examples have been given throughout this thesis) and to a large extent GIS do meet this need, but only insofar as automating and facilitating the data handling. Although the need and responsibility of collecting the data and manipulating it into effective parameters for the model is one step removed from the model, these task must still be done well and the issues and problems identified by many authors (e.g. Beven, 1989; Reddi, 1990; Grondin et al., 1990) still apply.

8.5.2 Surrogate data, error and accuracy

Throughout this project there have been many examples where surrogate data has had to be used to estimate the required data; e.g. semivariogram and cross-semivariogram modelling (Section 5.7); remote sensing (Section 7.4), soil particle size distributions (Section 6.6) etc. Using surrogate data as opposed to directly observed measurements to generate the parameters for a model, results in a loss of accuracy of the parameters. Or rather increases the uncertainty of the parameters and thus increases the uncertainty of the model output.

Error is a perennial concern both in GIS and environmental modelling, and has been discussed in many papers. Goodchild (1993) described an ideal GIS which would track error through the system, and "accuracy would a be feature of every product generated by the GIS". If loosely linked process models are to be fitted into Goodchild's (1993) system then error estimates would need to be part of the input and output of each model. However, most models, including Modflow and the others being evaluated, do not explicitly give a measure of error, and, current GIS do not meet this ideal.

This project uses several models together, each will have an associated innate error as well as errors in the input data. To what extent these errors add up or cancel out when the models are run together remains to be tested. Until general error quantification techniques are developed little can be done except compare model outputs with observed data to get an estimate of accuracy. This also has problems, the calibration process of a model for a particular area or problem is an attempt to minimise error under those particular conditions. The final measure of error gained

from such a calibration process cannot be assumed to be the innate error of the modelling system when applied to other areas.

Because of the way the system has been designed it is perhaps easier to handle some of these issues than if a complex, but user friendly GUI had been used. It was assumed from the start (Section 3.5.1) that the model links would be embedded within some higher level controlling process. This process could just as easily be designed to undertake a Monte-Carlo assessment of uncertainty or a calibration procedure (as in Section 5.8.2) as controlling the flow of data between several models as originally intended.

However, these are issues that are very important and that have yet to be satisfactorily addressed; considerable further work is still need in these areas.

8.6 Evaluation

The initial aim of the project has not been fully achieved, however some useful conclusions can be drawn. In the literature it was found that only the work of Vermulst et al. (1996) followed the idea of linking a soil moisture model to a groundwater model in the way illustrated in Figure 3.11. The work of Vermulst et al. (1996) was performed at a much smaller scale (nation-wide) than attempted here. The unsaturated zone was modelled on the basis that conditions were homogeneous at a $500 \times 500m$ pixel scale. The whole model system is used for assessing the effects of national water use policy on, for example, desiccation, eutrophication of surface waters etc. The models on which the work of Vermulst et al. is based have significant conceptual differences to those used here, e.g. the unsaturated flow model was not finite element but had a more conceptual¹ basis.

Returning to the theme of synthesis that was begun in Section 1, this project has attempted to bring together many different disciplines to solve a problem. Many of the techniques, studies, models and data on which this project draws have to some extent been developed in isolation. Although often the researchers had an idea that their work would have some practical use within some larger framework, there is often an implicit assumption that the other parts of the framework, whether other models, other sets of data or whatever would be of good quality. In reality this is not the case, in this project good models (i.e. models that, if well set up and given good data, will give good results) have had to be parameterised with bad or at least unknown quality data.

With respect to the original aim of the project much work still needs to be done. All aspects of surrogate data discussed in this project still require considerable research; statistics and geo-statistics for watertable prediction, classification of remote sensing using other data as additional bands in the classification, soil particle size distributions obtained from general soil descriptions for generating input parameters for models etc. It is also recommended that much simpler models are used in any future modelling, for example though SWMS_2D has considerable potential for modelling the water moisture profile, given the time for setting it up, especially gathering the data

¹ conceptual in the hydrological model sense, see Section 3.2.

it would have been better in retrospect to use the simpler approach of van Keulen (1986) to model the whole soil moisture and crop system.

Chapter 9

Conclusions

The original aim of the project was to develop a modelling system to assess and quantify the impact of groundwater extraction on crop production. Constrained by cost and time, the strategy was one of; (i) using existing components wherever possible, i.e. a standard GIS as a medium with which to link existing environmental process models; (ii) adopting a approach of loose linkage and changing the components as little as possible; and (iii) using secondary and surrogate data in preference to gathering primary data. As research progressed the project evolved to a more philosophical examination of the problems and issues that this strategy implied. These issues were concerned with the following areas;

- the tools and utilities required to achieve the model–GIS linkage;
- controlling the operation and flow of data between different environmental subsystem models to simulate a whole environmental super-system;
- reconciliation of the data-models used by the different process models and by the GIS;
- generation of adequate parameter sets for the models, particularly when surrogate and secondary data is used.

The constraints necessitated a loose or data-level linkage approach. However such an approach has advantages as well as disadvantages compared with a tightly integrated approach. The advantages are that a loose linked approach is relatively cheap and quick, ideal as a prototype, but it can form a finished modelling system in its own right or be part of the long term and changing system of a corporation. There is flexibility as the components can be recombined within different controlling processes for different purposes. The disadvantages are that the resulting system can be less efficient, less elegant and less user-friendly, though this is not necessarily the case — the linkage can be made transparent to the user and hidden behind a friendly interface (that can be constructed at a later date when the system has proved itself).

9.1 Specific recommendations and conclusions

Each major section in this thesis has ended with an evaluation of the research undertaken, in this section the conclusions and recommendations are brought together and grouped in a coherent and more logical form. These conclusions concern; the use of environmental process models; the linkage of process models and GIS; and the challenge presented by meeting the data need of the system.

9.1.1 Choosing and using environmental process models

The project has involved examining and using several environmental process models, with and without being linked to GIS, thus useful conclusions can be drawn about choosing and using them;

- a basic understanding of the assumptions and theory behind a model are crucial to ensure and be confident that it is used properly;
- before using a process model, a conceptual model of the relationships and interaction of all the components must be built. The conceptual model, which may be no more than clear idea in the mind, is then encoded into the process model;
- use the simplest model possible consistent with the available data and the task;
- use regression equations with care, tables of parameters are very robust and less influenced by anomalous situations.

To these conclusions some recommendations to model developers can be added;

- documentation should include many sample input files (there is no upper limit on the number that can be usefully included);
- documentation should include tables of default values and the limits of certain parameters;
- documentation should include an example modelling exercise plan which takes the user from development of a conceptual model through the first model runs, refining input parameters, sensitivity analysis and calibration to the final model run;
- error checking in the code should pick up bad input data and report the problem adequately.

9.1.2 Linking process models and GIS

This project involved using several environmental models which were either linked, already linked or implemented within the GIS; those that were linked were Modflow and SWMS_2D; a model already linked was *r.watershed* (Westervelt et al., 1993) and models implemented included van

Keulen and van Laar's (1986) crop model and of course Hedges's (1989) original cartographic modelling technique, in addition to many short cartographic modelling exercises that might not even be recognised as such.

The experience gained when linking SWMS_2D and Modflow to the GIS have led to the following conclusions:

- the choice between a loose linked at one extreme and a closely integrated approach to linkage/implementing such models within a GIS environment depends on; end user expertise, the size of the user base, resources, time schedules, and the importance or worth of the decisions being made;
- a loose linked approach can form the first prototype towards a more integrated system, it can be an end product or it can be a stage in the life cycle of a modelling system within an organisation;
- a command line interface to the model link enables it to be easily embedded within some higher level controlling process. This does not mean the model can not have an interactive interface as well;
- reconciling the data models used by the GIS and process models is a major concern. It may be desirable to change or enhance the data model of the GIS, e.g. by adding temporal objects as discussed in Section 8.3.2, but this is not necessary to achieve linkage and it should be justified if undertaken.

In Chapter 4 cartographic modelling techniques following the procedure of Hedges (1989) were undertaken within a standard GIS. On the basis of this work and on the numerous examples of cartographic modelling performed throughout the project it was concluded that:

- as a tool for planners Hedges's (1989) technique was practical and adequate to assess and quantify the effects of groundwater abstraction on crop production;
- a GIS proved a good environment to undertake this kind of work as it provided the functionality of Hedges's (1989) software as well additional functionality for data capture, handling manipulation, display and output generation;
- cartographic modelling proved useful both for preprocessing and post-processing data to and from the linked process models. For example in Section 5.8 *r.mapcalc* was used to subtract Modflow output from interpolated watertable level to highlight the differences; in Section 5.5.2 it was used to highlight anomalous artesian areas and in Section 6.7.2 it was used to generate stochastic realisation of soil properties.

On the subject of implementing models within a GIS, based on implementing *r.crop* the following conclusions can be made:

- when the model is well suited to the tools in hand, as van Keulen and van Laar's (1986) model was compatible with *r.mapcalc* then implementation within the GIS environment offers the advantages of; low overheads both in development and operation with regard to changing data formats, error checking and file handling;
- the theoretical models underlying the codes SWMS_2D and Modflow were not well suited to existing GRASS facilities, thus it would have been very difficult to implement a Modflow kind of model within *r.mapcalc* for example, and such an implementation would have been very slow. The model could have been implemented from scratch at a lower level in C using the GRASS libraries but this would have entailed considerable unnecessary effort and the loose linked approach was deemed more appropriate.

9.1.3 Data

The data for modelling has been a central focus of the project. Geostatistical, stochastic and remote sensing techniques were used to manipulate primary, secondary and surrogate data. Several conclusions can be drawn:

- if the data is not adequate a simpler model should be used even if the objectives of the modelling have to be altered to allow for this;
- the use of a GIS greatly facilitated the generation of model input parameters; functionality such as digitising, image process and cartographic modelling were all applied;
- once stochastic functions had been built into the GIS, stochastic datasets could be easily generated;
- geostatistical techniques have a great potential for helping to generate spatially varying parameter sets for models. This will be more fully realised when such techniques are fully integrated with GIS and when other knowledge of the surface can be included in the interpolation.

9.1.4 Conclusion

To summarise the work presented in this thesis, the linkage of the current generation of GIS and process models offers many advantages and raises many issues and challenges. This project has explored many of these and though inevitably as many new areas of research have been opened as resolved, the project has contributed significantly to the evolution of linked GIS-model systems, currently one of the prime concerns of the those involved in the managing of spatial data. In addition, it has allowed the logic of the design philosophy behind such systems to be explored and recommendations for a rationalised approach to be made based upon the intended end use and users.

References

- Aho, A. V., Kernighan, B. W. and Wienberger, P. J. (1988). *The AWK Programming Language*, Addison-Wesley Publishing Company.
- Allewijn, R. (1988). Regional hydrological systems analysis using satellite remote sensing data and a geographical information system: Application to groundwater modelling to the Roermond area, The Netherlands, *International Journal of Remote Sensing* 9: 1775–1785.
- Ashley, R. P. (1994). Examples of groundwater modelling in environmental assessment, *JIWEM* 8: 635–645.
- Avison, D. E. (1992). *Information Systems Development: a Database Approach*, Information Systems Series, 2nd edn, Blackwell Scientific Publications, Oxford, UK.
- Barry, R. G. (1976). Evaporation and transpiration, in R. J. Chorley (ed.), *Introduction to Physical Hydrology*, Methuen & Co Ltd, chapter 4.1.
- Bellmann, K., Ebert, W., Freier, B., Kunel, K., Matthaus, E., Schultz, A., and Vwenzel (1986). *Agroecosystem Modelling and Simulation - the Winter Wheat Agroecosystem Model AGROSIM-W*, Akademie der Landwirtschaftswissenschaften der Deutschen Demokratischen Republik, pp. 5–28.
- Belmans, C., Wesseling, J. G. and Feddes, R. A. (1983). Simulation model of the water balance of a cropped soil: SWATRE, *Journal of Hydrology* 63: 271–286.
- Beven, K. J. (1989). Changing ideas in hydrology – the case of physically-based models, *Journal of Hydrology* 105: 157–172.
- Bian, L., Sun, H., Blodgett, C., Egbert, S., Li, W., Ran, L. and Koussis, A. (1996). An integrated interface system to couple the SWAT model and Arc-Info, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').

- Booth, T. H. (1996). Predicting plant growth: Where will it grow? how well will it grow?, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Bouman, C. and Shapiro, M. (1992). Multispectral image segmentation using a multiscale image model, *Proceedings of the IEEE, International Conference on Acoustics, Speech and Signal Processing*, 23rd–26th March 1992, San Francisco, California, USA, pp. III.565–III.568.
- Bourne, S. R. (1983). *The UNIX System*, International Computer Science Series, Addison-Wesley Publishing Company.
- Bouwer, H. (1978). *Groundwater Hydrology*, Mc-Graw-Hill.
- Brooks, R. H. and Corey, A. T. (1964). Hydraulic properties of porous media, *Hydrology paper 2*, Colombia State University.
- Brooks, R. H. and Corey, A. T. (1966). Properties of porous media affecting fluid flow, *Proceedings of the ASCE, Journal of Irrigation Division* 92: 61–88.
- Burrough, P. A. (1986). *Principles of Geographical Information Systems for Land Resources Assessment*, number 12 in *Monographs on Soil and Resources Survey*, Clarendon Press, Oxford, UK.
- Chapra, S. C. and Canale, R. P. (1985). *Numerical Methods for Engineers*, Applied Mathematics Series, McGraw-Hill Book Company.
- Charnock, T. W., Elgy, J. and Hedges, P. D. (1996). Application of GIS linked hydrological models over a large area, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Charnock, T. W., Hedges, P. D. and Elgy, J. (1996). Linking multiple process level models to GIS, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 29–36.
- Charnock, T. W., Snoxell, J. and Elgy, J. (1994). Geographic information systems and digital mapping for water supply inventory, in C. Maksimović, F. Calomino and J. Snoxell (eds),

Water Supply Projects: New Technologies, NATO Advanced Study Institute (ASI), Partnership sub-series 2. Environment, Vol 15, 24th October – 4th November, Albena, Varna, Bulgaria, Springer Verlag, Berlin, Heidelberg, New York, pp. 537–552.

- Chiang, W. H. and Kinzelbach, W. (1992). *Processing Modflow: Pre- and Postprocessors for Simulation of Flow and Contaminants Transport in Groundwater Systems with MODFLOW, MODPATH and MT3D*, Hamburg, Heidelberg. User manual.
- Childs, E. C. (1969). *An Introduction to the Physical Basis of Soil Water Phenomena*, John Wiley and Sons Ltd.
- Childs, S. W., Gilley, J. R. and Splinter, W. E. (1977). A simplified model of corn growth under moisture stress, *Transactions of the American Society of Agricultural Engineers* 20(5): 858–865.
- Clark, I. (1979). *Practical Geostatistics*, Applied Science Publishers, Barking, UK.
- Clark, M. J. (1993). Data constraints on GIS application development for water resource management, in K. Kovar and H. P. Nachtnebel (eds), *HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources*, Publication No. 211, April 1993, Vienna, Austria, IAHS press, Wallingford, UK, pp. 451–463.
- Clarke, R. T. (1994). *Statistical Modelling in Hydrology*, John Wiley & Sons.
- Clemente, R. S., De Jong, R., Hayhoe, H. N., Reynolds, W. D. and Hares, M. (1994). Testing and comparison of three unsaturated soil water flow models, *Agricultural Water Management* 25: 135–122.
- Collins, F. C. and Bolstad, P. V. (1996). A comparison of spatial interpolation techniques in temperature estimation, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Cressie, N. A. C. (1993). *Statistics of Spatial data*, Wiley, New York, Chichester. Revised edition.
- Crompton, E. and Osmond, D. A. (1954). *The Soils of the Wem District of Shropshire, [Sheet 138]*, Memoirs of the Soil Survey of Great Britain, Ministry of Agriculture and Fisheries, Her Majesty's Stationary Office.
- D'Agnese, F., Faunt, C. C. and Turner, A. K. (1996). Using remote sensing and GIS techniques to estimate discharge and recharge fluxes for the Death Valley regional groundwater flow system, USA, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems*

- in *Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 503–511.
- de Wit, C. T. and Penning de Vries, F. W. T. (1985). Predictive models in agricultural production, *Philosophical Transactions of the Royal Society of London B* 310: 308–315.
- Deutsch, C. V. and Journel, A. G. (1992). *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, Oxford, UK.
- DeWeist, R. J. M. (1968). *Geohydrology*, John Wiley & Sons Inc, New York, USA.
- Dixon, A. J., Bradford, R. B., Cooper, D. M., Reeve, C. E. and Tucker, D. K. (1989). A hydro-geological assessment of the impact of proposed sand and gravel extraction on an adjacent ecologically sensitive floodplain environment — a case study, *Proceedings of the 6th Extractive Industry Geology Conference*, University of Birmingham, Birmingham, UK, pp. 70–81.
- Doucet, P. and Sloep, P. D. (1992). *Mathematical Modelling in the Life Sciences*, Ellis Horwood Series in Mathematics and its Applications, Ellis Horwood, London, UK.
- Dowd, P. A. (1994). *Mineral resource and environmental geostatistics: geostatistical simulation*, University of Leeds. Course manual.
- Dugas, W. A. and Ainsworth, C. G. (1985). Effect of potential evapotranspiration estimates on crop model simulations, *Transactions of the ASAE* 28(2): 471–475.
- Elgy, J., Charnock, T. W. and Hedges, P. D. (1994). Relevant GIS and remote sensing technology: Theoretical considerations, in C. Maksimović, F. Calomino and J. Snoxell (eds), *Water Supply Projects: New Technologies*, NATO Advanced Study Institute (ASI), Partnership sub-series 2. Environment, Vol 15, 24th October - 4th November, Albena, Varna, Black Sea, Bulgaria, Springer Verlag, Berlin, Heidelberg, New York, pp. 467–489.
- Elgy, J., Maksimović, C. and Prodanovic, D. (1993). Matching standard GIS packages with urban storm drainage simulation software, in K. Kovar and H. P. Nachtnebel (eds), *HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management*, Publication No. 211, April 1993, Vienna, Austria, IAHS press, Wallingford, UK.
- Ellis, F. (1996). The application of machine learning techniques to erosion modelling, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').

- Engels, B., Navular, K., Cooper, B. and Hahn, L. (1996). Estimating groundwater vulnerability to nonpoint source pollution from nitrates and pesticides on a regional scale, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 521–533.
- EPA (1992). Groundwater modeling compendium model fact sheets, descriptions applications and assessment framework, *Fact sheets, Report No. EPA/500/B-92/006*, Environment Protection Agency (EPA), Office of Solid Waste and Emergency Response, EPA, Washington DC., USA.
- Evans, D. A. (1992). *Development of a Conceptual Model of the Soil moisture*, PhD thesis, Aston University, Birmingham, UK.
- Feddes, R. A., Kowalik, P. J. and Zaradny, H. (1978). *Simulation of field water use and crop yield*, Simulation Monographs, PUDOC, Wageningen, Holland.
- Fedra, K. (1993). GIS and environmental modelling, in M. F. Goodchild, B. O. Parks and L. T. Steyaert (eds), *Proceedings of the Second International Conference on Integrated Geographic Information Systems and Environmental Modeling*, September 1993, Breckenridge, Colorado, USA, Oxford University Press, Oxford, UK, pp. 35–50.
- Fels, J. E. and Matson, K. C. (1996). A cognitively-based approach hydrogeomorphic land classification using digital terrain models, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Fisher, T. R. (1993). Use of 3d geographic information systems in hazardous waste site investigations, in M. F. Goodchild, B. O. Parks and L. T. Steyaert (eds), *Proceedings of the Second International Conference on Integrated Geographic Information Systems and Environmental Modeling*, September 1993, Breckenridge, Colorado, USA, Oxford University Press, Oxford, UK, pp. 239–247.
- Gessler, P., McKenzie, N. and Hutchinson, M. (1996). Progress in soil-landscape modelling and spatial prediction of soil attributes for environmental models, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').

- Goodchild, M. F. (1993). Data models and data quality, problems and prospects., in M. F. Goodchild, B. O. Parks and L. T. Steyaert (eds), *Proceedings of the Second International Conference on Integrated Geographic Information Systems and Environmental Modeling*, September 1993, Breckenridge, Colorado, USA, Oxford University Press, Oxford, UK, pp. 94–103.
- Goodchild, M. F., Parks, B. O. and Steyaert, L. T. (eds) (1996). *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Gray, F. E. G. (1980). Inspectors report to the Secretary of State for the Environment following the Inquiry into the Shropshire Groundwater Scheme.
- Grindley, J. (1967). The estimation of soil moisture deficits, *The Meteorological Magazine* 96(1137).
- Grondin, G. H., Gannett, M., van der Heijde, P. K. M. and Patt, R. O. (1990). Critical errors that hydrogeologic professionals can make with computer programs, *Transferring Models to Users, Symposium Proceedings*, American Water Resources Association (AWRA), Denver, USA.
- Hackett, C. (1991). *Plantgro : a software package for the coarse prediction of plant growth*, CSIRO, Melbourne, Australia.
- Hanks, R. J. and Hill, R. W. (1980). *Modeling Crop Responses to Irrigation: In relation to soils, climate and salinity*, International Irrigation Information Centre. IIIC publications; no 6. Distributed by Pergamon, Oxford, UK.
- Harris, J., Gupta, S., Woodside, G. and Ziemba, N. (1993). Integrated use of a GIS and a three-dimensional, finite-element model: San Gabriel Basin groundwater flow analyses, in M. F. Goodchild, B. O. Parks and L. T. Steyaert (eds), *Proceedings of the Second International Conference on Integrated Geographic Information Systems and Environmental Modeling*, September 1993, Breckenridge, Colorado, USA, Oxford University Press, Oxford, UK, pp. 168–172.
- Hayhoe, H. N. and De Jong, R. (1982). *Computer simulation of soil water movement and uptake by plant root*, Publication Number 82-13, Land Resource Research Institute, Ottawa, Ontario, Canada.
- Hazelton, N. W. J. (1991). Extending GIS to include dynamic modelling, *Proceedings of the 3rd Colloquium of the Spatial Research Centre*, May 1991, University of Otago, Dunedin, New Zealand, pp. 73–82.

- Hedges, P. D. (1989). *The Impact on Agriculture of the Drawdown of Shallow Watertables*, PhD thesis, Aston University, Birmingham, UK.
- Hedges, P. D. and Walley, W. J. (1983). A study of soil moisture losses due to the drawdown of a shallow water table, *Scientific Procedures Applied to the Planning, Design and Management of Water Resources Systems, Proceedings of the Hamburg Symposium*, Publication no. 147, August 1983, Hamburg, Germany, IAHS, Wallingford, UK, pp. 489–499.
- Hempel, C. G. (1965). *Aspects of Scientific Explanation: and Other Essays in the Philosophy of Science*, Collier-Macmillan, London, UK.
- Hilton, K. (1986). *Process and Pattern in Physical Geography*, Bell and Hyman.
- Hobbs, W. W. (1993). *RDB: A Relational Database Management System*. RDB is a free set of relational database operators, available via the internet via ftp at 'sunsite.doc.ic.ac.uk' in the directory '/Mirrors/ftp.funet.fi/pub/languages/perl/CPAN/modules/dbperl/scripts/rdb/' in file 'RDB-2.5.tar.Z' which includes a postscript version of Hobb's documentation.
- Hopmans, J. W. and Gutierrez-Rave, E. (1988). Calibration of a root water uptake model in spatially variable soils, *Journal of Hydrology* 103: 53–65.
- Ireland, P. (1995). GIS: Taking root in a hostile climate, *GIS Europe* 4(6): 24–27.
- Isaaks, E. H. and Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics*, Oxford University Press, Oxford, UK.
- Jenny (1980). *The Soil Resource*, number 37 in *Ecological Studies*, Springer Verlag, Berlin, Germany.
- Jensen, M. E. (1968). Water consumption by agricultural plants, in T. T. Kozlowski (ed.), *Water Deficits and Plant Growth*, Vol. 2, Academic Press, New York, USA.
- Jones, C. A. and Kiniry, J. R. (1986). *CERES-Maize*, Texas A&M University Press, Texas, USA.
- Jones, R. J. A. (1975). Soils in Staffordshire II, sheet SJ 82 (Eccleshall), *Map sheet and memoir*, Soil Survey of England and Wales, Harpenden, UK.
- Kamps, P. T. W. J. and Oltshoorn, T. N. (1996). Using GIS for hierarchical refinement of mod-flow models, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 535–550.

- Kesteven, J. and Hutchinson, M. (1996). Spatial modelling of climatic variables on a continental scale, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Knipe, C. V., Lloyd, J. W., Lerner, D. N. and Greswell, R. (1993). *Rising Groundwater Levels in Birmingham and the Engineering Implications*, number 92 in *CIRIA Special Publication*, Construction Industry Research and Information Association, 6 Storey's Gate, Westminster, London SW1P 3AU.
- Kool, J. B. and van Genuchten, M. T. (1991). HYDRUS: One-dimensional variable saturated flow and transport model, including hysteresis and root water uptake, *Research Report 124*, US Department of Agriculture, Agriculture Research Service (ARS) Salinity Laboratory, Riverside, California, USA.
- Kopp, S. M. (1996). Linking GIS and hydrological models: where have we been, where are we going?, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 133–139.
- Kovar, K. and Nachtnebel, H. P. (eds) (1996). *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK.
- Kowalczyk, A. and Kemp, A. (1993). Incorporating the temporal dimension in a GIS application, *GIS Research UK, Proceedings of the First National Conference*, 18th–20th March 1993, University of Keele, Staffordshire, UK, pp. 46–57.
- Laurini, R. and Thompson, D. (1995). *Fundamentals of Spatial Information Systems*, APIC, Academic Press.
- Liaw, A. and Crawford, R. (1995). *gnuplot 3.5 User's Guide*, Department of Statistics, Texas A&M University.
- Ludwig, B., Daroussin, J., King, D. and Souchère, V. (1996). Using GIS to predict concentrated flow erosion in cultivated catchments, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 429–436.

- Maas, S. J. and Doraiswamy, P. C. (1996). Integration of satellite data and model simulations in a GIS for monitoring regional evaporation and biomass production, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- MAFF (1977). *Sandland Irrigation 1977 Edition*, Gleadthorpe Experimental Husbandry Farm, ADAS, Ministry of Agriculture Fisheries and Food, Mansfield, UK.
- Maguire, D. J. and Dangermond, J. (1991). *The Functionality of GIS*, Vol. 1, Longman, London, UK, pp. 319–333.
- Majure, J. J., Cressie, N., Cook, D. and Symanzik, J. (1996). GIS, spatial statistical graphics, and forest health, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Marshall, T. J. and Holmes, J. W. (1979). *Soil Physics*, Cambridge University Press, Cambridge, UK.
- Matson, K. C. and Fels, J. E. (1996). Approaches to automated water table mapping, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- McDonald, M. G. and Harbaugh, A. W. (1988). *A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model*, United States Geological Survey. Modeling Techniques Book 6, Chapter A1.
- Meijerink, A. M. J., Van Lieshout, A. M. and Rahnema Mobareke, F. (1996). Comparison of approaches for erosion modelling using flow accumulation with GIS, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 437–444.
- Meteorological Office (1981). The Meteorological Office rainfall and evaporation calculation system (MORECS), *Hydrological Memorandum 41*, Meteorological Office, UK.

- Meteorological Office (1995). MORECS version 2, rainfall evaporation and soil moisture data for environmental resource management, *Publicity material*, Meteorological Office.
- Michl, C. (1996). Using GIS Modflow and Modpath for groundwater management of an alluvial aquifer of the River Sieg, Germany., in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 551–557.
- Mitasova, H. and Hofierka, L. (1993). Interpolation by regularized spline with tension: I, *Mathematical Geology* **25**: 641–655.
- Musgrave, D. C. (1981). Letter to the Chief Executive, Severn Trent Water Authority, referring to the Shropshire Groundwater Scheme Public Local Inquiry, Department of the Environment, 29th May 1981.
- NRA (1985). Shropshire Groundwater Scheme, report on the 1984 test pumping and regulation, *Internal report*, National Rivers Authority.
- NRA (1989). Shropshire Groundwater Scheme, regulation pumping: 1989, *Internal report*, National Rivers Authority.
- NRA (1994). Shropshire groundwater scheme, third phase: Leaton area.
- OS (1993). Land-Form Panorama: NTF v2.0, contours level 1, DTM level 5, *Temporary format guide*, Ordnance Survey, Southampton, UK.
- Pannatier, Y. (1994). Variowin 2 (geostatistics software package) help file. Demonstration versions of Variowin are available via the internet, '<http://www-sst.unil.ch/Software.html>'.
- Pears, N. (1985). *Basic Biogeography*, 2nd edn, Longman, Harlow, UK.
- Peuquet, D. J. (1984). A conceptual framework and comparison of spatial data models, *Cartographica* **21**: 66–113.
- Price, M., Morris, B. and Robertson, A. (1982). A study of intergranular and fissure permeability in chalk and Permian aquifers, using double-packer injection testing, *Journal of Hydrology* **54**: 401–423.
- Ragg, J. M., Beard, G. R., George, H., Heaven, F. W., Hollis, J. M., Jones, R. J. A., Palmer, R. C., Reeve, M. J., Robson, J. D. and Whitfield, W. A. D. (1984). *Soils and Their Use in Midland and Western England*, Bulletin No. 12, Soil Survey of England and Wales, Harpenden, Herts, UK.

- Ramlal, B. and Beard, K. (1996). An alternate paradigm for representing soils data and data quality information, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Reddi, L. N. (1990). Potential pitfalls in using groundwater models, *Transferring Models to Users, Symposium Proceedings*, American Water Resources Association (AWRA), Denver, USA, pp. 131–141.
- Rippon, P. W. and Wyness, A. J. (1994). Integrated catchment modelling as a water resources management tool, *JIWEM* 8: 671–679.
- Ritjema, P. E. (1965). An analysis of actual evapotranspiration, *Agriculture research report*, Pudoc, Wageningen, Holland.
- Robinson, G. W. (1943). Mapping the soil of Britain, *Discovery* 4: 118–121.
- Rogowski, A. S. and Engman, E. T. (1996). Using a using a SAR image and a decision support system to model spatial distribution of soil water in a GIS framework, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Ross, M. A., Fielland, C. E. and Tara, P. D. (1990). An integrated GIS/hydrologic model for phosphate mining reclamation design, *Transferring Models to Users, Symposium Proceedings*, American Water Resources Association (AWRA), Denver, USA, pp. 31–39.
- Ross, M. A., Schwartz, M. A. and Tara, P. D. (1991). Groundwater modelling with GIS: An example, *Proceedings of the International Symposium on Ground Water in Practice*, ASCE Hydraulics Division, Nashville, Tennessee, USA, pp. 244–249.
- Ross, M. A. and Tara, P. D. (1993). Integrated hydrologic modeling with geographic information systems, *Journal of Water Resources Planning and Management* 119(2): 129–140.
- Rushton, K. R. (1996). Pers comm, september 1996.
- Rushton, K. R. and Bishop, T. J. (1993). Water resource study of the Nottinghamshire Sherwood sandstone aquifer of eastern England: Mathematical model of the Sherwood sandstone aquifer, *Report*, Department of Civil Engineering, The University of Birmingham, Birmingham, UK.

- Rushton, K. R. and Redshaw, S. C. (1979). *Seepage and Groundwater Flow*, John Wiley & Sons.
- Salama, R. B., Laslett, D. and Farrington, P. (1993). Predictive modelling of management options for the control of dryland salinity in a first-order catchment in the wheatbelt of Western Australia, *Journal of Hydrology* 145: 19–40.
- Savabi, M. R., Klik, A., Grulich, K., Mitchell, J. K. and Nearing, M. A. (1996). Application of WEPP and GIS on small watersheds in USA and Austria, in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 469–476.
- Schädlich, S. and Mauser, W. (1996). Spatial evapotranspiration calculation on a microscale test site using the GIS-based PROMET., in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 649–657.
- Senn, J. A. (1990). *Information Systems in Management*, 4th edn, Wadsworth Publishing Co., Belmont California, USA.
- Shapiro, M. and Westervelt, J. (1991). r.mapcalc an algebra for GIS and image processing, *Technical report*, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois 61801, U.S.A.
- Shapiro, M., Westervelt, J., Gerdes, D., Larson, M. and Brownfield, K. R. (1993). *GRASS 4.1 Programmer's Manual*, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois 61801, U.S.A.
- Sharp, R. G. (1978). Paper presented to the Midland branch of the Institution of Civil Engineers, 30th november.
- Shaw, E. (1993). *Hydrology in Practice*, 2nd edn, Chapman and Hall, London, UK.
- Simunek, J., Vogel, T. and van Genuchten, M. T. (1994). *The SWMS_2D Code for Simulating Water Flow and Solute Transport in Two-Dimensional Variably Saturated Media*, US Salinity Laboratory, Agriculture Research Service (ARS), US Department of Agriculture, Riverside, California, USA.
- Skinner, A. C. (1979). Proof of evidence of A.C. Skinner: Shropshire Groundwater Scheme, Public Local Inquiry, Shrewsbury, Severn Trent Water Authority.
- Soil Survey of England and Wales (1982). Shropshire Groundwater Scheme, soil and hydrological survey, *Report*, Rothamsted Experimental Station, Harpenden, Herts, UK.

- Sophocleous, M. (1992). Groundwater recharge estimation and regionalization: the Great Bend Prairie of central Kansas and its recharge statistics, *Journal of Hydrology* 137: 113–140.
- SRA (1971). Shropshire groundwater investigation, *Report 1*, Severn River Authority.
- SRA (1974). Shropshire groundwater investigation, *Report 2*, Severn River Authority.
- Stewart, J. I. and Hagan, R. M. (1973). Functions to predict effects of crop water deficits, *Journal of the Irrigation and Drainage Division of ASCE* 100: 179–199.
- STWA (1975). Shropshire groundwater investigation, *Report 3*, Severn Trent Water Authority.
- STWA (1976). Shropshire groundwater investigation, *Report 4*, Severn Trent Water Authority.
- STWA (1977). Shropshire groundwater, a report of the investigation and proposals for development, *Report*, Severn Trent Water Authority.
- STWA (1978). Shropshire groundwater investigation, *Report 5*, Severn Trent Water Authority.
- Thapa, K. (1992). Accuracy of spatial data used in geographic information systems, *Photogrammetric Engineering and Remote Sensing* 58(6): 835–841.
- Tomlin (1991). *Cartographic Modelling*, Vol. 1, Longman, London, pp. 361–374.
- Tscheschke, P. D. and Gilley, J. R. (1979). Status and verification of Nebraska's corn crop growth model — CORNGRO, *Transactions of the American Society of Agricultural Engineers* 22(6): 1329–1337.
- Turner, A. K., Ervin, E. M. and Downey, J. S. (1991). Evaluation of geographic information systems for three-dimensional ground-water model, Yucca Mountains Nevada, *Proceedings of the 2nd Annual International Conference on High Level Radioactive Waste Management*, pp. 520–528.
- USDA (1995). Water erosion prediction project, hillslope profile and watershed model documentation, *Report 10*, United States Department of Agriculture, Agricultural Research Service (ARS), National Soil Erosion Research Laboratory (NSERL), West Lafayette, Indiana, USA. (edited by D C Flanagan and M A Nearing).
- van Genuchten, M. T. (1980). A closed form equation for predicting the hydraulic conductivity of unsaturated soils, *American Journal of the Soil Science Society* 44: 892–898.
- van Keulen, H. (1986). A simple model of water-limited production, in H. van Keulen and J. Wolf (eds), *Modelling of Agricultural Production: Weather Soil and Crops*, Simulation Monographs, Pudoc, Wageningen, Holland, chapter 3.4.

- van Keulen, H. and van Laar, H. H. (1986). The relation between water use and crop production, in H. van Keulen and J. Wolf (eds), *Modelling of Agricultural Production: Weather Soil and Crops*, Simulation Monographs, Pudoc, Wageningen, Holland, chapter 3.3.
- van Keulen, H. and Wolf, J. (eds) (1986). *Modelling of Agricultural Production: Weather Soil and Crops*, Simulation Monographs, Pudoc, Wageningen, Holland.
- Vaughan, P., Simunek, J., Suarez, D. L., Corwin, D. L. and Rhoades, J. D. (1996). Mapping CO₂ surface flux in an irrigated agricultural area, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NC-GIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa.fe.html').
- Vermulst, J. A. P. H., Hoogeveen, J., Lange, W. D., Bos, H. B. and Pakes, U. (1996). MONA: an interface for GIS-based coupled saturated and unsaturated groundwater modelling in The Netherlands., in K. Kovar and H. P. Nachtnebel (eds), *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of HydroGIS'96*, Publication No. 235, 16th–19th April 1996, Vienna, Austria, IAHS press, Wallingford, UK, pp. 575–583.
- Vieux, B. E., Farajalla, N. S. and Gaur, N. (1993). Integrated GIS and distributed stormwater runoff modeling, in M. F. Goodchild, B. O. Parks and L. T. Steyaert (eds), *Proceedings of the Second International Conference on Integrated Geographic Information Systems and Environmental Modeling*, September 1993, Breckenridge, Colorado, USA, Oxford University Press, Oxford, UK.
- Wachowicz, M. and Healey, R. G. (1993). Towards temporality in GIS, *GIS Research UK, Proceedings of the First National Conference*, 18th–20th March 1993, University of Keele, Staffordshire, UK, pp. 58–68.
- Wagenet, R. J. and Hutson, J. L. (1989). *LEACHM: Leaching Estimation and Chemistry Model. Version 2*, Center for Environmental Research, Cornell University, Ithaca and New York.
- Wall, L., Christiansen, T. and Schwartz, R. L. (1996). *Programming PERL*, 2nd edn, O'Reilly.
- Walley, M. J. and Hussein, D. A. (1982). Development and testing of a general purpose soil-moisture-plant model, *Hydrological Sciences* 27(1): 1–17.
- Walley, W. J. (1979). Proof of evidence of W.J. Walley: Shropshire Groundwater Scheme, Public Local Inquiry, Shrewbury.
- Walley, W. J. and Hedges, P. D. (1979). The effect of groundwater drawdown upon available soil moisture. Department of Civil Engineering, University of Aston in Birmingham, UK.

- Westervelt, J., Gerdes, D., Larson, M. and Brownfield, K. R. (1993). *GRASS 4.1 User's Manual*, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois 61801, U.S.A.
- Wiesner, C. J. (1970). *Climate, Irrigation and Agriculture*, Angus & Robertson (UK) Ltd.
- Wilson, J. P. (1996). GIS-based surface/subsurface modeling: New potential for new models, *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 21st–25th January 1997, Santa Fe, New Mexico, USA, National Centre for Geographic Information Analysis (NCGIA), University of California, Santa Barbara, California, USA, (on CD ROM and available via internet 'http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html').
- Zepp, H. and Belz, A. (1992). Sensitivity and problems in modelling soil moisture conditions, *Journal of Hydrology* **131**: 227–238.

Appendix A

The code

More than ten thousand lines of code were written during the course of the project, these are split amongst large applications, trivial programs, and modification and debugging exercises on existing code. In this section some of the more significant and interesting bits of code and those referred to in the main text are included.

Before the documenting the code a brief summary of programming under GRASS and UNIX is provided to explain the files and scripts included in this Appendix. The first section is a general overview of developing applications within GRASS. The second section documents the *r.crop* implementation of van Keulen and van Laar's (1986) model. The third section discusses modifying *r.mapcalc* and includes some modifications that were necessary for this project. The fourth section looks at PERL and GRASS and the fifth section discusses the *RDB* database which was used for managing the watertable data supplied by the NRA.

A.1 Writing programs within GRASS

Many of the programs written for this project and described below were written using the facilities provided by GRASS. Effectively GRASS creates a programming environment which is a customisation of the UNIX environment and handles much of the administrative tasks associated with writing and maintaining code, as well as providing many useful libraries of functions. As in UNIX there are two general levels on which applications can be developed, at a low level programs can be written in C and compiled together with UNIX and GRASS libraries to produce fast binary executable programs. At a higher level scripts can be written that are sequence of shell commands, these can call low level C programs, and other scripts to perform tasks. The use and philosophy of shell scripts is discussed in Section 3.4.3. It should be noted that C programs can issue commands to the shell through a *system call*, the command could initiate a second program a script or perform some other administrative task.

A.1.1 Writing GRASS C programs

The main system development language for UNIX is C, and GRASS is also written in C. C is a procedural language similar to FORTRAN or Pascal. It is a compiled language, i.e. before the computer can understand it, the source code (the instructions written by the programmer in a text file) must be converted into binary or executable code by a program called a compiler. Compilation in C is a three stage process:

1. preprocessing: each text file that makes up the program (usually identified by a *.c* extension) is examined. At the time any header files (identified by a *.h* extension) specified in the *.c* file are appended. Header files contain standard definitions for data types, constants and functions, the UNIX system provides some header files which programmers can use (e.g. such as *stdio.h* which includes definitions for the C input and output functions). GRASS also provides header files which can be included if required in programs;
2. compilation: each *.c* file is converted to a system specific binary *object* file (identified by a *.obj* extension).
3. linkage: the object files are linked together with any libraries specified into a single executable piece of code. UNIX provides many system libraries and GRASS provides several more.

***Gmakefile* files**

Every program written in the GRASS environment has a *Gmakefile* this is similar to a regular UNIX *makefile*. In a *makefile* the interdependencies between different files of code which make up a program are specified. Thus when a piece of code is changed all other pieces of code on which it depends and which depend on it are also recompiled. This is initiated by typing the UNIX command *make*. *makefiles* can be terse and hard to write, however *Gmakefile* were designed to be simpler and the program which initiates the recompilation (*Gmake4.1*) knows where all the GRASS libraries, source code, binary code and include files are located. For further information the reader is directed to the GRASS programming manual (Shapiro et al., 1993) and to the numerous UNIX books available (e.g. Bourne, 1983).

The VASK library

There are several libraries supplied with GRASS, these include libraries to handle display, raster files, vector structures, site files etc. A library that was used extensively for the Modflow link, particularly for the *mod.edit* program is the *visual ask* or VASK library. This is used for constructing form interfaces, a typical piece of code from the *mod.edit* program is as follows (for further information on the VASK library the user is referred to the GRASS manual (Shapiro et al., 1993)):

the user can simply type the command line and then is prompted for the required information. For example the program *r.sample* has the following syntax if used in command line mode:

```
r.sample [-s]1 map=name coords=east,north[,east,north,...] [mapset=name]
```

This program takes one *flag*; -s, and three options; map, coords and mapset. Note also the the option coords can be provided with more than one value and this value is split into two parts (an *x* and *y* coordinate pair).

If the user just types the command name then a question and answer session is initiated:

```

GRASS 4.1 > r.sample

OPTION: existing raster
key: map
required: YES

Enter the name of an existing raster file
Enter 'list' for a list of existing raster files
Hit RETURN to cancel request
> dem
<dem>

OPTION: coords to sample
key: coords
format: east,north
required: YES
multiple: YES
enter option > 358500,325000

You have chosen:
coords=358500,325000
Is this correct? (y/n) [y]
enter option > 359000,325500

You have chosen:
coords=359000,325500
Is this correct? (y/n) [y]
enter option >

OPTION: import to search
key: import
required: NO
enter option >

FLAG Set the following flag?
out in site format?(y/n) [n] n
79 90

```

The parser allows the programmer considerable scope for constraining the input. The programmer can decide which command line options are “optional” and which the user must provide. The type of arguments that must be provided is specified, e.g. the argument might be a raster name in which case the parser will check the raster exists and in the interactive mode will give the user the chance to list out the available rasters. The range of the argument can be constrained, e.g. a floating point number greater than 0 and smaller than one. The parser also allows the programmers own error checking functions to be included.

The source code to initiate the *r.sample* commandline options is as follows:

¹ Square brackets '[']' are standard notation for optional arguments.

```

#include <gis.h>
#include <gisdefs.h>

struct Option* map;
struct Option* coords;
struct Option* mapset;
struct Flag* sites;
struct Cell_head region;

/*****
sample a rast map at a point or list of points and print the value
found to stdout, handles fp and nulls
*****/
main(argc,argv)
int argc;
char* argv[];
{
    int fp;
    struct Cell_head region;
    int i,j,k;
    double x,y;
    int col,row;
    DCELL* dcell;
    char* try_mapset;
    char* search_mapset;

/*initiatate this is a GRASS program*/

    G_gisinit(argv[0]);

/*define options*/

    map=G_define_option();
    coords=G_define_option();
    mapset=G_define_option();
    sites=G_define_flag();

/*characterise options*/

    map->key="map";
    map->required=YES;
    map->description="existing raster";
    map->type=TYPE_STRING;
    map->gisprompt="mapset,cell,raster";

    coords->key="coords";
    coords->key_desc="east,north";
    coords->required=YES;
    coords->multiple=YES;
    coords->description="coords to sample";
    coords->type=TYPE_DOUBLE;

    mapset->key="mapset";
    mapset->required=NO;
    mapset->description="mapset to search";
    mapset->type=TYPE_STRING;

    sites->key='s';
    sites->description="out in site format";

/*call the parser*/

    if(G_parser(argc,argv))
    {
        G_fatal_error("parse failed: main()");
    }
    ...
    ...
}

```

For further information the reader is referred to the GRASS programming manual (Shapiro et al., 1993).

A.1.2 Writing GRASS shell scripts

Section A.1.1 explained that the GRASS parser allowed the options for a program to be specified on the command line. The principal reason for this is that it allows these programs to be embedded within a shell script. As discussed in Section 3.4.3 it is the shell that interpretes and executes the users command when written on the command line or specified in a shell script. Many shell scripts have been developed for UNIX, but perhaps the most ubiquitous are the the C-shell and the Bourne shell. Rather than create a whole new shell, the GRASS developers simply extend the functionality provided by whatever shell the user normally uses. This is done by; setting environment variables — which hold important information, such as the location of programs and libraries — that the shell can access; and by providing some providing programs that look and

act like shell commands.

A shell script can be a series of commands which might just as easily be type at the command prompt. For example the commands type in sequence at the GRASS prompt, e.g.

```

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > g.region rast=dem

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > d.mon start=x0
Graphics driver [x0] started

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > d.rast dem

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > d.rast soil

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > d.rast drift_type

Mapset <PERMANENT> in Location <shropshire>
GRASS 4.1 > d.mon stop=x0
Monitor 'x0' terminated

```

could be placed into a text file

```

#!/bin/sh

g.region rast=dem
d.mon start=x0
d.rast dem
d.rast soil
d.rast drift_type
d.mon stop=x0

```

Both alternatives will start a GRASS monitor, display three rasters in it and then close the monitor again. This is the simplest form of shell script, but shell scripts allow all the important programming structures such as loops and selection statements.

A good example of a shell script application is the *r.section* script. *r.section* is a shell script which combines GRASS commands, AWK scripts and other utilities such as the familiar UNIX function plotting package *Gnuplot*, to produce an interactive facility for drawing cross-sections. The user specifies which rasters should be used and picks two points on the current GRASS monitor. *r.section* then joins the two point with a line on the monitor, plots the cross sections in a *Gnuplot* monitor and optionally writes the data to a file (Figure A.1). The GRASS programs that are used to do this are:

- *g.tempfile*, to generate temporary scratch files;
- *g.gisenv*, to set up correct environment variables;
- *g.findfile*, to check specified rasters exist;
- *d.where*, to get coordinates from the mouse;
- *d.mapgraph*, to draw the line on the selected monitor;
- *r.profile*, to extract the profile data.

The watertable cross-sections of Figures 5.14 and 5.22 were generated using this program.

```

#!/bin/sh

#check this is a grass program
if test "$GISRC" = ""
then
    echo "Run this program in GRASS">&2
    exit 1
fi

```

```

eval 'g.gisenv' : ${GISSET} ${GISDBASET} ${LOCATION_NAME} ${MAPSET}

temp1='g.tempfile pid=$$'
temp2='g.tempfile pid=$$'
temp3='g.tempfile pid=$$'

case $$ in
0)
    echo usage: r.section \map_layer1 map_layer2 ..\ output_name {yrange\{x:y\}}
;;
1)
    echo usage: r.section \map_layer1 map_layer2 ..\ output_name {yrange\{x:y\}}
;;
*)

#check that raster layers are valid

for i in $1
do
    eval 'g.findfile element=cell file=$i'
    if [ ! "$file" ]
    then
        echo raster $i does not exist
        exit
    fi
done

#get start and finish coordinates

echo
echo
echo GET START COORDINATES WITH THE MOUSE

d.where > $temp1

start_coors='awk '
{
    east=$1
    north=$2
}
END
{
    printf("%s,%s",east,north)
} ' $temp1 '

if [ $start_coors = , ]
then
    echo no coordinates chosen
    exit
fi

echo
echo
echo GET END COORDINATES WITH THE MOUSE

d.where > $temp1

end_coors='awk '
{
    east=$1
    north=$2
}
END
{
    printf("%s,%s",east,north)
} ' $temp1 '

if [ $end_coors = , ]
then
    echo no coordinates chosen
    exit
fi

#draw transect on the screen

echo move $start_coors | sed 's/,/ /g' >$temp1
echo draw $end_coors | sed 's/,/ /g' >>$temp1
d.mapgraph input=$temp1 color=black
cp $temp1 $2.mg

#sort out the gnuplot instruction file

echo $1 > $temp1

number_list='awk '
{
    i=1
    while (i <= (NF - 1)) {
        printf("%d ",i+1)
        i++
    }
} ' $temp1 '

last_numbers='awk '
{
    printf("%d",NF+1)
} ' $temp1 '

```

```

if test $# -ge 3
then
    echo set xrange[$3] > $2.gp
    echo plot \> >> $2.gp
else
    echo plot \> > $2.gp
fi

for i in $number_list
do
    title='awk '
    {
        printf("%s", $('i'-1))
    } $temp1 '

    echo \'$2.dat\' using 1:$i title \'$title\' with lines, \>> $2.gp
done

title='awk '
{
    printf("%s", $NF)
} ' $temp1 '

echo \'$2.dat\' using 1:$last_number title \'$title\' with lines >> $2.gp
echo pause -1 \'hit return to remove plot\' >> $2.gp

#sort out the gnuplot data file

for i in $1
do
    r.profile map=$i line=$start_coords,$end_coords | sed 's/ /\n/g' >>$temp2
done

#work out the distance between points

echo $start_coords > $temp1
echo $end_coords >> $temp1

distance='awk '
BEGIN {
    FS=","
    count=1
}
count==1 {
    startx=$1
    starty=$2
}
count==2 {
    stopx=$1
    stopy=$2
}
{
    count++
}
END {
    dx=startx-stopx
    dy=starty-stopy
    distance=sqrt((dx+dx)+(dy+dy))
    printf("%s",distance)
} ' $temp1 '

units='awk '
BEGIN {
    count=1
}
count==1 {
    nos_points=$1
}
{
    count++
}
END {
    dist=""$distance"
    printf("%lf",('$distance'/(nos_points-1)))
} ' $temp2 '

awk '
BEGIN {
    count=1
}
count==1 {
    nos_points=$1
}
count > 1 && count <= (nos_points-1) {
    row[count-1]=row[count-1] " " $1
}
{
    count=count+1
}
count==(nos_points +2) {
    count=1
}
END {
    count=1
}

```



```

while (count <= nos_points)
{
    printf("%d %s\n",count * '@units',row[count])
    count++
}
printf("\n")
} ' $temp2 >> $temp3

cp $temp3 $2.dat

#clean up

rm $temp1
rm $temp2
rm $temp3

#plot the cross sections

gnuplot $2.gp

esac

```

The script *r.crop* (see Appendix A.2) is another example of a shell script. For further information the reader is referred to the GRASS programming manual and the GRASS user manual (Shapiro et al., 1993; Westervelt et al., 1993).

A.2 The *r.crop* program

r.crop is an implementation of van Keulen and van Laar's (1986) model, it has been written using *r.mapcalc* statements embedded within a Bourne shell script, see Section 7.5.

```

#!/bin/sh
#check this is a grass program

if test "$GISRC" = ""
then
    echo "Run this program in GRASS">&2
    exit 1
fi

eval 'g.gisenv' : ${GISBASE?} ${GISDBASE?} ${LOCATION_NAME?} ${MAPSET?}

temp1='g.tempfile pid=$$'

case $$ in
2)
    #REQUIRE INPUT RASTERS:
    $PDr.constant potential rooting depth (mm)
    $SMo.constant pore space or saturate moisture content (cm^3/cm^3)
    $SMfc.constant field capacity (cm^3/cm^3)
    $SMwp.constant wilting point (cm^3/cm^3)
    $a.constant values 'a' parameter in conductivity equation (cm^2/d)
    $alp.constant values 'alpha' parameter in conductivity equation (cm^-1)
    $hmax.constant values of 'maximum h' parameter in conductivity equation (cm)
    $Ko.constant saturated conductivity (cm d^-1)
    $Vr.initial moisture in root zone (mm)
    $Vnr.initial moisture in non root zone (mm)
    $RD.initial initial root depth (mm)
    $WRt.initial initial dry weight of roots (kg/ha)
    $WLV.initial initial dry weight of leaves (kg/ha)
    $WST.initial initial dry weight of stems (kg/ha)
    $WGR.initial initial dry weight of grain (kg/ha)
    $Ta.1-n time-series of average temperatures (deg C)
    $P.1-n time-series of precipitation (mm)
    $Ie.1-n time-series of effective irrigation (mm)
    $Pgs.1-n time-series of pot' gross CO2 assimilation (kg/ha/day)
    $ETo.1-n time-series of total atmos' evaporation demand, Panman (mm/day)
    $crop.type crop type (1= wheat)
    $soil.type soil type

    #DATA TABLES
    p_atts="2, 9,5"
    ptable="0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0, 1,2,3,4,5,\
0.45,0.38,0.30,0.25,0.23,0.20,0.18,0.16,0.15,\
0.60,0.50,0.43,0.35,0.30,0.28,0.25,0.23,0.20,\
0.75,0.65,0.55,0.45,0.40,0.38,0.33,0.30,0.25,\
0.88,0.75,0.65,0.55,0.50,0.48,0.43,0.38,0.35,\
0.92,0.85,0.75,0.65,0.60,0.55,0.50,0.48,0.45"

    partition_atts="2, 4,12"
    partition_table="1,2,3,4, 0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,2.0, \
0.50,0.50,0.00,0.00,\
0.42,0.58,0.00,0.00,\
0.33,0.67,0.00,0.00,\
0.26,0.80,0.14,0.00,\
0.20,0.52,0.28,0.00,\
0.15,0.45,0.42,0.00,\
0.10,0.34,0.56,0.00,\
0.07,0.23,0.70,0.00,\

```



```

STSM      number of degree days so far
STSM total total degree days for all completed growth stages
TREQ      the degree days required for full growth
Ta        average temperature of timestep(degC)
DVS a     development stage of end of 'a' sub-timestep
DVS b     development stage of end of 'b' sub-timestep
Stimestop length of sub-timestep 'a'
Stimestop length of sub-timestep 'b'
Stagea    growth stage of sub-timestep 'a'
Stageb    growth stage of sub-timestep 'b'

r.mupcalc "TSM Stime=TSM $lasttime + (Ta $time + Stimestop)"

r.mupcalc "DVS Stime=eval(
TREQ=lookupa $crop_atts,stagea,1,crop_type,$crop_table), \
TSM $time/TREQ "

r.mupcalc "stageb=if(DVS $time>stagea+1,stageb+1,stageb)"
r.mupcalc "TSM total=if(DVS $time>stagea+1,TSM total+ \
lookupa($crop_atts,stagea,1,crop_type,$crop_table))"

r.mupcalc "DVS $time =if(DVS $time>stagea+1,stagea+1,DVS $time)"
r.mupcalc "DVS $time b=eval( \
TREQ=lookupa $crop_atts,stageb,1,crop_type,$crop_table), \
if(DVS $time>stageb+1,(TSM $time-TSM total)/TREQ+stageb,DVS $time )"

r.mupcalc "timestop=eval( \
TREQ=lookupa($crop_atts,stagea,1,crop_type,$crop_table), \
if(DVS $time<1.0,Stimestop, \
round((TREQ-TSM $lasttime)/Ta $time)))"

r.mupcalc "timestopb=Stimestop-timestopa"

# I N E A

#-----Root Depth-----
RDR      root depth (mm)
RGR      is root growth rate (12mm/day)

r.mupcalc "RD $time =eval( \
R=lookupa $crop_atts,stagea,2,crop_type,$crop_table), \
RD.$lasttime+(R*timestopa))"

#-----Maximum Evaporation-----
SEa      maximum evaporation (mm/day)
SEa      a reduction factor due to incomplete radiation interception
SETo     total atmospheric evaporative demand, Pmmmm (mm/day)

r.mupcalc "SE $time =eval( \
Th=1-exp(-0.6*LAI.max), \
SETo $time*(1-Th))"

#-----Actual Evaporation-----
SEa      actual evaporation (mm/day)
SEa      maximum evaporation (mm/day)
SESR      volumetric soil moisture of root zone (cm3/cm3)
SESRfc    is field capacity (cm3/cm3)
SESRa     air dry soil (cm3/cm3)

r.mupcalc "Ea $time =eval( \
SEa=0.333*SETo constant, \
Ea $time =((SESR.$lasttime-SEa)/(SESRfc.constant-SEa)))"

#-----Maximum Transpiration-----
TEa      maximum transpiration (mm/day)
TEa      a reduction factor due to incomplete radiation interception
LAI      leaf area index (m2/m2)
SETo     total atmospheric evaporative demand, Pmmmm (mm/day)

r.mupcalc "Te $time =eval( \
Th=1-exp(-0.6*LAI.$lasttime), \
SETo $time*Th)"

#-----Actual Transpiration-----
TEa      actual transpiration (mm/day)
SEa      maximum evaporation (mm/day)
Sp        is an empirical parameter from table
SESRcr    critical soil moisture below which roots extract suboptimally
S        (cm3/cm3)
SESRfc    is field capacity (cm3/cm3)
SESRwp    is wilting point (cm3/cm3)
SESR      volumetric soil moisture of root zone (cm3/cm3)
TEa      maximum transpiration (mm/day)

r.mupcalc "T.$time.=eval( \
p=lookupa($p_atts,SETo.$time/Stimestop,4,$ptable), \
SESRcr=(1-p)*(SESRfc.constant-SESRwp.constant)+ \
SESRwp constant, \
T=if(SESR.$lasttime>SESRcr,Te $time.a, \
((SESR.$lasttime-SESRwp constant)/ \
(SESRcr-SESRwp constant))*Te.$time.a))"

#-----Capillary rise/Drainage-----

```



```

%CR_B rate of capillary rise or drainage (mm/day)
%Rrd total head (mm)

r.mscalc "CR_B $time s=eval(
depth=UTD $time-SD $lasttime,
CR=10*(lookup($capillary_atta,depth/10,$r $lasttime/10,
$capillary_table)),
Rrd=$r $lasttime-depth,
if(Rrd,CR,0,$r $lasttime))"

#-----Soil Moisture in Root Zone-----

%Rur thickness of non-rooted part of root zone (mm)
%RPr potential root zone depth (mm)
%RD root depth (mm)

r.mscalc "Rur $time s=%RPr constant-SD $lasttime"

%RRr amount of moisture added due to root growth (mm/day)
%Rur moisture content of non-rooted zone (mm)
%Rr root growth rate (mm/day)
%Rur thickness of non-rooted part of root zone (mm)

r.mscalc "RRr $time s=eval(
Rr=lookup($crop_atta,stages,2,$crop_type,$crop_table),
Rur $lasttime=%Rr $time a)"

%RRr rate of change of soil moisture (mm/day)
%P precipitation in timetop (mm)
%RRr amount of moisture added due to plant growth (mm/day)
%Ie irrigation in timetop (mm)
%T actual transpiration (mm/day)
%RRr amount of moisture added due to plant growth (mm/day)
%Ea actual evaporation (mm/day)
Assuming all the input are evenly spread over the 'a' period

r.mscalc "RRr $time s=(P $time+Ie $time)/timetop+
RRr $time a + CR_B $time a - Ea.$time.a - T $time.a)"

%Rr total soil moisture in the root zone (mm)
%RRr rate of change of soil moisture (mm/day)

r.mscalc "Rr $time s=%Rr $lasttime + (%RRr $time.a *timetop)"

%SRr volumetric soil moisture of root zone (cm3/cm3)
%Rr total soil moisture in the root zone (mm)
%RD root depth (mm)

r.mscalc "SRr $time s=Rr $time a/%RD $time.a"

#-----Matrix Head and Conductivity-----

%SR - matrix head in the root zone (cm)
%SRr volumetric moisture content in root zone (cm3/cm3)
By empirical parameter

r.mscalc "Sr $time s=eval(
exp(sqrt(-log(SRr $time a/SRr constant)/By)))"

%Kx conductivity (mm/day)
%SR matrix head of root zone (cm)
%Sa empirical value (22.6)

r.mscalc "Kx $time s=if (Sr $time s<=Sa constant,
Sa constant*exp 2.718282,-a/p constant*Sr $time.a),
a constant*exp(Sr $time a,-1.4))"

#-----Moisture in Non-rooted Zone-----

%Rur moisture in non rooted zone (mm)
%Rur depth of non-rooted zone (mm)
%RPr potential rooting depth of crop (mm)
%RD root depth (mm)
%RRr amount of moisture added due to plant growth (mm/day)
%Rr root growth rate

r.mscalc "Rur $time s=eval(
Rur=%RPr constant-SD $lasttime,
Rr=lookup($crop_atta,stages,2,$crop_type,$crop_table),
RRr=%Rr $lasttime=%Rr $time,
Rur $lasttime=RRr*timetop)"

#-----Gross Assimilation-----

%GCASS actual gross assimilation taking into account water shortage (kg/ha/d)
%Fh reduction factor due to incomplete interception of radiation
%Pgs potential gross CO2 assimilation (kg/ha/day)
%PGASS potential gross assimilation with no water stress (kg/ha/day)
%LAI leaf area index (m2/m2)
%T actual transpiration (mm/day)
%Tn maximum transpiration (mm/day)

r.mscalc "GCASS $time s=eval(
Fh=1-exp(-0.6*LAI $lasttime),
PGASS=Fgs $time,
PGASS=(T $time a/Tn $time a))"

```

```

$-----Rate of Increase of Dry Weight-----
#DMI rate of increase of dry weight of vegetation (kg/ha/day)
#ASAG assimilates available for growth (kg/ha/day)
#MRES maintenance respiration (kg/ha/day)
#RRpa pre-anthesis phase respiration rate, (0.015kgCH2O/kg/day)
#Ce efficiency associated with converting primary photosynthates to
$ plant structural material 70%

RRpa=0.015
Ce=0.7
r.mapcalc "DMI.$time.a=eval(
MRES=$RRpa*TVDL.$lasttime,
ASAG=GA55.$time.a-MRES,
Ce=ASAG )"

$-----Dry Weight of Roots-----
#VRT dry weight of roots (kg/ha)
#DS average development stage of period
#FR root partitioning factor
#IVRT rate of increase of dry weight of root system (kg/ha/day)
#DMI rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "VRT.$time.a=eval(
DS=(DVS.$lasttime+DVS.$time.a)/2,
FR=lookupn($partition_atts,crop_type,DS,$partition_table),
IVRT=FR*DMI.$time.a,
VRT.$lasttime+(IVRT*timestep))"

$-----Dry Weight of Leaves-----
#VLV dry weight of leaf blades kg/ha
#DS average development stage of timestep
#DWLV rate of decline of leaves due to water shortage (kg/ha/day)
#FL partitioning factor for leaves
#IVLV rate of increase of dry weight of leaves (kg/ha/day)
#DMI rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "IVLV.$time.a=eval(
DS=(DVS.$lasttime+DVS.$time.a)/2,
FL=lookupn($partition_atts,2,DS,$partition_table),
FL=DMI.$time.a)"

r.mapcalc "DWLV.$time.a=eval(
0.03*(1-T.$time.a/Tm.$time.a)*VLV.$lasttime)"

r.mapcalc "VLV.$time.a=eval(
VLV.$lasttime+((IVLV.$time.a-DWLV.$time.a)*timestep))"

$-----Dry Weight of Stems-----
#VST dry weight of stems kg/ha
#DS average development stage of timestep
#FS partitioning factor for stems
#IVST rate of increase of dry weight of stems (kg/ha/day)
#DMI rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "VST.$time.a=eval(
DS=(DVS.$lasttime+DVS.$time.a)/2,
FS=lookupn($partition_atts,3,DS,$partition_table),
IVST=FS*DMI.$time.a,
VST.$lasttime+(IVST*timestep))"

$-----Dry Weight of Grain-----
#WGR dry weight of grain kg/ha
#DS average development stage of timestep
#FG partitioning factor for grain
#IWGR rate of increase of dry weight of grain (kg/ha/day)
#DMI rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "WGR.$time.a=eval(
DS=(DVS.$lasttime+DVS.$time.a)/2,
FG=lookupn($partition_atts,4,DS,$partition_table),
IWGR=FG*DMI.$time.a,
WGR.$lasttime+(IWGR*timestep))"

$-----Leaf Area Index LAI-----
SLIA leaf area index (m^2/m^2)
#CSLA constant specific leaf area (20m^2/kg)
#VLV dry weight of leaves

CSLA=20
r.mapcalc "LAI.$time.a=VLV.$time.a*CSLA/10000"
r.mapcalc "LAI.max=if(LAI.max>LAI.$time.a,LAI.max,LAI.$time.a)"

$-----Total Dry Weight-----
#TVDL total dry weight (kg/ha)
#TVLD total live dry weight (kg/ha)
#TVDL total dead dry weight (kg/ha)

r.mapcalc "TVDL.$time.a=VRT.$time.a + VLV.$time.a + VST.$time.a + WGR.$time.a"
r.mapcalc "TVDL.$time.a=TVDL.$lasttime+DWLV.$time.a*timestep"
r.mapcalc "TVL.$time.a=TVDL.$time.a + TVDL.$time.a"

$ I N E $
$-----Root Depth-----
#RD root depth (mm)
#RR is root growth rate (12mm/day)

r.mapcalc "RD.$time.a=eval(
RR=lookupn($crop_atts,stageb,2,crop_type,$crop_table),

```

```

RD.$time.a+(Rr*$timestepb))"

#-----Maximum Evaporation-----
$Em      maximum evaporation (mm/day)
$fh      a reduction factor due to incomplete radiation interception
$ETo     total atmospheric evaporative demand, Panman (mm/day)

r.mapcalc "Em.$time=eval(
fh=1-exp(-0.6*LAI.max),
ETo.$time*(1-fh))"

#-----Actual Evaporation-----
$Ea      actual evaporation (mm/day)
$Em      maximum evaporation (mm/day)
$SHr     volumetric soil moisture of root zone (cm^3/cm^3)
$SHfc    is field capacity (cm^3/cm^3)
$SHA     air dry soil (cm^3/cm^3)

r.mapcalc "Ea.$time=eval(SHa=0.333*$Nwp.constant,
Ea.$time.a+((SHr.$time.a-$Ha)/(SHfc.constant-$Ha)))"

#-----Maximum Transpiration-----
$Tm      maximum transpiration (mm/day)
$fh      a reduction factor due to incomplete radiation interception
$LAI     leaf area index (m^2/m^2)
$ETo     total atmospheric evaporative demand, Panman (mm/day)

r.mapcalc "Tm.$time=eval(
fh=1-exp(-0.6*LAI.$time.a),
ETo.$time*$fh)"

#-----Actual Transpiration-----
$T       actual transpiration (mm/day)
$Em      maximum evaporation (mm/day)
$P       is an empirical parameter from table
$SHcr    critical soil moisture below which roots extract suboptimally
$        (cm^3/cm^3)
$SHfc    is field capacity (cm^3/cm^3)
$SHwp    is wilting point (cm^3/cm^3)
$SHr     volumetric soil moisture of root zone (cm^3/cm^3)
$Tm      maximum transpiration (mm/day)

r.mapcalc "T.$time=eval(
p=lookup($p_atts,$To.$time/$timestep,4,$ptable),
SHcr=(1-p)*(SHfc.constant-$Nwp.constant)+
$Nwp.constant,
T=(($SHr.$time.a-$SHcr)*$Tm.$time,
((($SHr.$lasttime-$Nwp.constant)/
($SHcr-$Nwp.constant))*$Tm.$time))"

#-----Capillary rise/Drainage-----
$CR_D    rate of capillary rise or drainage (mm/day)
$Hrd     total head (mm)

r.mapcalc "CR_D.$time=eval(
depth=$Td.$time-$D.$time.a,
CR=10*(lookup($capillary_atts,depth/10,$r.$time.a/10,
$capillary_table)),
Hrd=$r.$time.a-(depth),
if(Hrd,CR,0,$r.$time.a))"

#-----Soil Moisture in Root Zone-----
$Dnr     thickness of non-rooted part of root zone (mm)
$PDr     potential root zone depth (mm)
$RD      root depth (mm)

r.mapcalc "Dnr.$time=$PDr.constant-$D.$time.a"

$Dnr     amount of moisture added due to root growth (mm/day)
$Vnr     moisture content of non-rooted zone (mm)
$Rr      root growth rate (mm/day)
$Dnr     thickness of non-rooted part of root zone (mm)

r.mapcalc "dRr.$time=eval(
Rr=lookup($crop_atts,$stageb,2,$crop_type,$crop_table),
Vnr.$lasttime*$Rr/$Dnr.$time)"

$dRr     rate of change of soil moisture (mm/day)
$P       precipitation in timestep (mm)
$dRr     amount of moisture added due to plant growth (mm/day)
$Ie      Irrigation in timestep (mm)
$T       actual transpiration (mm/day)
$dRr     amount of moisture added due to plant growth (mm/day)
$Ea      actual evaporation (mm/day)
$        assuming that all the input from rain and irrigation
$        are added to the a period

r.mapcalc "dVr.$time=(dRr.$time - Ea.$time - T.$time) "

$Vr      total soil moisture in the root zone (mm)
$dVr     rate of change of soil moisture (mm/day)

r.mapcalc "Vr.$time=Vr.$time.a + (dVr.$time*$timestepb)"

$SHr     volumetri soil moisture of root zone (cm^3/cm^3)
$Vr      total soil moisture in the root zone (mm)
$RD      root depth (mm)

```



```

r.mapcalc "SHr.$time=Ur.$time/RD.$time"

$-----Matrix Head and Conductivity-----
SHr    matric head in the root zone (cm)
SHr    volumetric moisture content in root zone (cm3/cm3)
Sy      empirical parameter

r.mapcalc "Sr.$time=eval(
exp(sqrt(-log(SHr.$time/SHo.constant)/Sy)))" \

SHr    conductivity (um/day)
SHr    matric head of root zone (cm)
Sa      empirical value (22.6)

r.mapcalc "Tr.$time=if (Sr.$time<hmax.constant, \
h0.constant*exp(2.718282,-alp.constant*Sr.$time), \
a.constant*exp(Sr.$time,-1.4))" \

$-----Moisture in Non-rooted Zone-----
$Wur    moisture in non rooted zone (mm)
$Dwr    depth of non-rooted zone (mm)
$Dwr    potential rooting depth of crop (mm)
$RD      root depth (mm)
$SHr    amount of moisture added due to plant growth (mm/day)
$Gr      root growth rate

r.mapcalc "Wur.$time=eval(
Wur=Pr.constant-RD.$time.a, \
Rr=lookupn($crop_atts,stageb,2,crop_type,$crop_table), \
SHr=Wur.$time.a*Gr/Wur, \
Wur.$time.a-$SHr*timestop)" \

$-----Gross Assimilation-----
$GASS    actual gross assimilation taking into account water shortage (kg/ha/d)
$Ffr      reduction factor due to incomplete interception of radiation
$Fgsa     potential gross CO2 assimilation (kg/ha/day)
$FPGASS    potential gross assimilation with no water stress (kg/ha/day)
$LA1      leaf area index (m2/m2)
$T        actual transpiration (mm/day)
$Tm       maximum transpiration (mm/day)

r.mapcalc "GASS.$time=eval(
Ffr=exp(-0.6*LA1.$time.a), \
PGASS=ch*Fgs.$time, \
PGASS*(T.$time/Tm.$time))" \

$-----Rate of Increase of Dry Weight-----
$DWI      rate of increase of dry weight of vegetation (kg/ha/day)
$ASAG     assimilates available for growth (kg/ha/day)
$MRES     maintenance respiration (kg/ha/day)
$R2pa     pre-anthesis phase respiration rate, (0.015kgCH2O/kg/day)
$Cp       efficiency associated with converting primary photosynthates to
           plant structural material 70%

$R2pa=0.015
Cp=0.7

r.mapcalc "DWI.$time=eval(
MRES=$R2pa*Tm.$time.a, \
ASAG=$GASS.$time-MRES, \
$Cp*ASAG)" \

$-----Dry Weight of Roots-----
$WRT      dry weight of roots (kg/ha)
$DS       average development stage of period
$FR       root partitioning factor
$INRT     rate of increase of dry weight of root system (kg/ha/day)
$DWI      rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "WRT.$time=eval(
RD=(stageb+$FS.$time.b)/2, \
FR=lookupn($partition_atts,crop_type,$DS,$partition_table), \
INRT=FR*$DWI.$time.a, \
WRT.$time.a+(INRT*timestop))" \

$-----Dry Weight of Leaves-----
$WLV      dry weight of leaf blades kg/ha
$DS       average development stage of timastop
$DWLV     rate of decline of leaves due to water shortage (kg/ha/day)
$FL       partitioning factor for leaves
$DWLV     rate of increase of dry weight of leaves (kg/ha/day)
$DWI      rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "WLV.$time=eval(
RD=(stageb+$FS.$time.b)/2, \
FL=lookupn($partition_atts,2,$DS,$partition_table), \
FL*$DWI.$time)" \

r.mapcalc "DWLV.$time=eval(
0.03*(1-T.$time.a/Tm.$time.a)*WLV.$time.a)" \

r.mapcalc "WLV.$time=eval(
WLV.$time.a+((DWLV.$time-DWLV.$time)*timestop))" \

$-----Dry Weight of Stems-----
$WST      dry weight of stems kg/ha
$DS       average development stage of timastop
$FS       partitioning factor for stems
$INWST    rate of increase of dry weight of stems (kg/ha/day)
$DWI      rate of increase of dry weight of vegetation (kg/ha/day)

```

```

r.mapcalc "VST.$time=eval(
DS=(stageb+DVS.$time.b)/2,
PS=lookupn($partition_atts,5,DS,$partition_table),
IVST=PS*DNI.$time,
VST.$time.a=(IVST*timestep))"

#-----Dry Weight of Grain-----
#VGR dry weight of grain kg/ha
#DS average development stage of timestep
#PS partitioning factor for grain
#IVST rate of increase of dry weight of grain (kg/ha/day)
#DNI rate of increase of dry weight of vegetation (kg/ha/day)

r.mapcalc "WGR.$time=eval(
DS=(stageb+DVS.$time.b)/2,
PS=lookupn($partition_atts,6,DS,$partition_table),
IVGR=PS*DNI.$time,
WGR.$time.a=(IVGR*timestep))"

#-----Leaf Area Index LAI-----
#LIA leaf area index (m^2/m^2)
#CSLA constant specific leaf area (20m^2/kg)
#VLV dry weight of leaves

CSLA=20
r.mapcalc "LAI.$time=VLV.$time*CSLA/10000"
r.mapcalc "LAI.max=(if(LAI.max>LAI.$time,LAI.max,LAI.$time))"

#-----Total Dry Weight-----
#TVD total dry weight (kg/ha)
#TVL total live dry weight (kg/ha)
#TVD total dead dry weight (kg/ha)

r.mapcalc "TVL.$time=VET.$time + VLV.$time + VST.$time + WGR.$time"
r.mapcalc "TVD.$time=TVL.$time.+DVLV.$time*timestep"
r.mapcalc "TWD.$time=TVL.$time + TVD.$time"

#-----Remove unneeded rasters-----
g.remove "TSUN.$lasttime"
g.remove "DVS.$lasttime"
g.remove "DVS.$time.a"
g.remove "DVS.$time.b"
g.remove "RD.$lasttime"
g.remove "RD.$time.a"
g.remove "Ra.$time"
g.remove "Ra.$time.a"
g.remove "Ea.$time"
g.remove "Ea.$time.a"
g.remove "Ta.$time"
g.remove "Ta.$time.a"
g.remove "T.$time"
g.remove "T.$time.a"
g.remove "Dmr.$time"
g.remove "Dmr.$time.a"
g.remove "dMr.$time"
g.remove "dMr.$time.a"
g.remove "dMr.$time"
g.remove "dMr.$time.a"
g.remove "Mr.$lasttime"
g.remove "Mr.$time.a"
g.remove "MR.$lasttime"
g.remove "MR.$time.a"
g.remove "Tr.$lasttime"
g.remove "Tr.$time.a"
g.remove "Tr.$lasttime"
g.remove "Tr.$time.a"
g.remove "Mar.$lasttime"
g.remove "Mar.$time.a"
g.remove "QASS.$time"
g.remove "QASS.$time.a"
g.remove "DNI.$time"

g.remove "DNI.$time.a"
g.remove "VET.$time.a"
g.remove "VLV.$time.a"
g.remove "IVLV.$time"
g.remove "IVLV.$time.a"
g.remove "DVLV.$time"
g.remove "DVLV.$time.a"
g.remove "VST.$time.a"
g.remove "WGR.$time.a"
g.remove "TVL.$time.a"
g.remove "TVD.$time.a"
g.remove "TWD.$time.a"
g.remove "LAI.$lasttime"
g.remove "LAI.$time.a"

#-----Time and Stage-----
lasttime=$time
r.mapcalc "stagea=stageb"
done

#END OF TIMESTEP

#cleanup unnecessary rasters
g.remove "TSUN.$lasttime"
g.remove "DVS.$lasttime"
g.remove "RD.$lasttime"

```

```

g remove "Ur.$lasttime"
g remove "SUr.$lasttime"
g remove "Sr.$lasttime"
g remove "Er.$lasttime"
g remove "Uur.$lasttime"
g remove "LAI.$lasttime"
g remove "TSM total"
g remove "stages"
g remove "stagesh"
g remove "timesteps"
g remove "timesteph"
g remove "LAI max"
..
*)
echo "usage "
echo "r.crop timestep_length number_of_timesteps"
esac

```

A.3 Adapting *r.mapcalc*

The main raster manipulation or cartographic modelling (see Section 3.3.3) facility in GRASS is the program *r.mapcalc*. It is an implementation of a *map algebra* syntax, and allows operations similar to those described by Burrough (1986). For a full description of *r.mapcalc* see Shapiro and Westervelt (1991) and Shapiro et al. (1993), further capabilities were added to it with the introduction of floating point operations, see 3.5.2.

r.mapcalc follows to the design philosophy of GRASS in that it is design to be *open*, meaning that it is easy to improve and added to. For this project several new operations were added to GRASS, this section describes how this is done. As with the rest of GRASS, *r.mapcalc* is written in the 'C' programming language.

A.3.1 Adding new operations

To add a new operation to *r.mapcalc*, 'C' code functions must be added to the *r.mapcalc* source code. At least two new 'C' functions are required for each new operation added to *r.mapcalc*.

The first function simply tests whether a valid number of arguments have been passed to the new *r.mapcalc* operation, this generally has the form:

```
int n_operation_name(int n)
```

where *n* is the number of arguments passed to the function. The function returns *1* if this is acceptable or *0* otherwise.

The second function performs the calculation and has the form:

```
x_operation_name(double* argv, int argc, double* xcell, int ncols)
```

where *argv* is a two dimensional array of arguments, one set of arguments for each pixel of a row, *argc* is the number of arguments per pixel, *xcell* is a pointer to an array of doubles, which is where the return values are written, one for each pixel of a row, and *ncols* is the number of columns (or pixels per row).

An optional third function, of the form:

```
i_operation_name(int* argv, int argc, int* cell, int ncols)
```

performs the same task as the "*x*." function, but is purely integer based.

The function.h header file (found in the *r.mapcalc/mapcalc* directory) must be edited to include these new functions:


```

/***** FUNCTION DECLARATIONS *****/

int      x_nores(), x_nores(),
int i_small() x_small(), x_small ,
int i_small ,x_small(), x_small();

struct function_list function_list[]=
{
    "abs" i_abs, x_abs, n_abs, 1,
    "eval" i_eval, x_eval, n_eval, 1,
    "float", DEFUNC, x_2float, n_2float, 1,
    "if", i_if, x_if, n_if, 1,
    "int", i_2int, x_2int, n_2int, 0,
    "isnull" i_isnull,x_isnull, n_isnull, 0,
    "exp", DEFUNC, x_exp, n_exp, 1,

    "y" DEFUNC, x_y, n_y, 1,
    "nores" DEFUNC, x_nores, n_nores, 1,
    "nores", DEFUNC, x_nores, n_nores, 1,
    * add new functions above this line */
    "", DEFUNC, DEFUNC, DEFUNC, 0
}

```

The code file is added to the *Gmakefile*:

```

PCN = r.mapcalc
NAME=$(BIN_DIR)_CHD
LIST = main o\
      compare o\
      convert.o\
      evaluate o\
      execute o\
      expression o

      xsin o\
      xexp o\
      xtan o\
      xlookup o

LINES = $(CFILES) $(HFILES) $(ROUTINES)

$(NAME)/$(PCN) $(LIST) $(LINES)
rm -f a.out
$ CC $(CFLAGS) $(LIST) $(LINES) $(HFILES) $(HFILES)
mv a.out $@

$(LIST) glob.h
evaluate o function.h
function o function.h
main o function.h
pollish o function

```

Several new *r.mapcalc* operations were needed during the course of this project. Three table look-up operations, a normal random number generator, a function to perform Simpson's $\frac{1}{3}$ rule on the soil-moisture function and a general function for performing Simpson's $\frac{1}{3}$ rule on any equation.

A.3.2 look-up functions

Agricultural and crop yield models frequently require values from tables. A model described by van Keulen 1986) was implemented with *r.mapcalc* commands embedded within a Bourne shell script (see Section 7.5 and Appendix A.2). For this three *lookup* operations were encoded. The first was a simple one dimensional *lookup*, given a key to an index the corresponding value is returned Figure A.2.a. The second is a multi-dimensional *lookup*, given n keys to n indexes the corresponding value is returned, Figure A.2.b. The third is a reverse multidimensional *lookup*, given a value and $n - 1$ keys to n indexes the corresponding n 'th key is returned, Figure A.2.c. All these functions employ linear interpolation when no exact match is found.

Because of the way *r.mapcalc* is encoded it is necessary to pass the whole table to the function when the operation is called. The is easy to do in a script as the table can be specify as a *string* variable, (see Figure A.3):

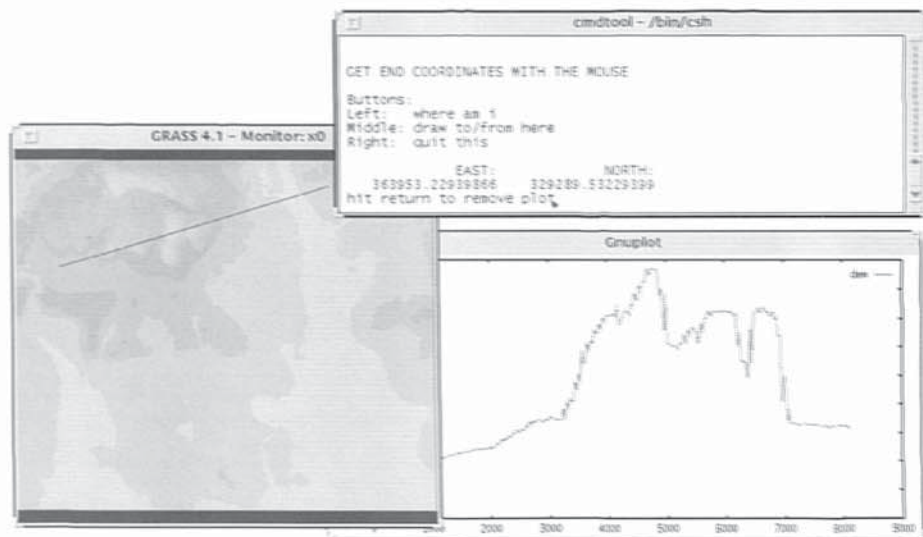


Figure A.1: Screen shot showing the operation of the *r.section* program.



Figure A.2: *look-up* functions in mapcalc, (a) a one dimensional *lookup* function, (b) an n-dimensional *lookup* function and (c) reverse n-dimensional *lookup* function

```

...
...
CR="0.02,0.06,0.10,0.15,0.20,0.30,0.4,0.50"
capillary_atts="2, 9,10"
capillary_table="\
0.0, 0.02, 0.06, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50,\
20,50,100,250,500,1000,2500,5000,10000,16000,\
20.1, 20.0, 19.9, 19.8, 19.7, 19.6, 19.3, 19.1, 18.9,\
49.8, 49.7, 49.1, 48.6, 47.9, 47.3, 46.0, 44.9, 43.8,\
98.7, 98.6, 91.1, 86.8, 82.4, 78.9, 73.3, 69.0, 65.4,\
165.8,165.5,126.6,111.2,100.1, 92.7, 83.0, 76.4, 71.5,\
218.4,218.3,147.3,124.0,108.8, 99.3, 87.4, 79.8, 74.2,\
264.8,264.7,163.8,134.1,115.6,104.4, 90.8, 83.3, 76.2,\
307.1,307.0,178.2,142.7,121.3,108.8, 93.7, 84.5, 78.0,\
332.2,332.1,186.7,147.8,124.7,111.3, 95.4, 85.8, 79.0,\
362.6,362.5,193.0,161.9,127.5,113.4, 96.8, 86.8, 79.8,\
363.5,363.4,197.1,164.1,128.9,114.4, 97.5, 87.3, 80.3"
...
...
#-----Capillary rise/Drainage-----
# CR_D rate of capillary rise or drainage (mm/day)
#Hrd total head (mm)

r.mapcalc "CR_D.$time.s=eval( \
depth=Vtd.$time-RD.$lasttime,\
CR=10*(lookup($capillary_atts,depth/10,$r.$lasttime/10,\
$capillary_table)),\
Hrd=$r.$lasttime-(depth),\
if(Hrd,CR,0,$r.$lasttime))"

```

Figure A.3: Passing a shell environment variable to an *r.mapcalc* function.

The code for the *lookupn* and *lookupr* functions follows:

```

#include "glob.h"

void* recurs_allocate(int n, int* list, int pos);
double recurs_linear_get(void* pointer,int n,int* list,int pos,double* query);
double recurs_reverse_get(void* pointer,int n,int* list,int pos,double* query,double key,double* index);

/*****
n_lookupn()
n-dimensional lookup function with linear interpolation
see also:
lookup() 1 dimensional lookup function
lookupr() reversed n dimensional lookup function

lookupn(number of dimensions (n),
length of dimension 1(l1),...length of dimension n(ln),
key 1,...,key2,
index(d1)1,...,index(d1)n,...,index(dn)1,...,index(dn)n,
value(d1)1,...,value(d1)l1,...,value(dn)1,...,value(dn)n)

a 1-d example

lookupn(1,5, 2.38, 0,1,2,3,4, 0,10,100,1000,10000)

mos dimensions | | | | |
length of dimension | | | | |
key into table | | | | |
index of table | | | | |
values in table | | | | |

returns 442

eg 0 1 2 3 4
0 10 100 1000 10000
442

a 2-d example (basic 2d table)
this table is table 20 from van Keulen and Volf(1986)
"Modelling of agricultural Production, Weather, Soil and Crops"
PUDOC Wageningen

lookupn(2, 9,5, 0 25,1.5, 0.2,0.3,0.4,0.6,0.6,0.7,0.8,0.9,1.0, 1,2,3,4,5,\
0.45,0.38,0.30,0.25,0.23,0.20,0.18,0.16,0.15,\
0.60,0.50,0.43,0.35,0.30,0.28,0.25,0.23,0.20,\
0.75,0.65,0.55,0.45,0.40,0.38,0.33,0.30,0.25,\

```



```

0.85,0.75,0.65,0.55,0.60,0.48,0.43,0.38,0.35,\
0.92,0.85,0.75,0.65,0.60,0.55,0.50,0.48,0.45)
returns 0.4825

eg
    0.25 --
    |
0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0
-----
1.5___ 1 0.45 0.38 0.30 0.25 0.23 0.20 0.18 0.16 0.15
2 0.60 0.50 0.43 0.35 0.30 0.28 0.25 0.23 0.20
3 0.75 0.65 0.55 0.45 0.40 0.38 0.33 0.30 0.25
4 0.85 0.75 0.65 0.55 0.50 0.48 0.43 0.38 0.35
5 0.92 0.85 0.75 0.65 0.60 0.55 0.50 0.48 0.45

a 3-d example (eg a more than one 2d table)

lookupn(3, 3,2,2, 0.06,50,2, 0.02,0.06,0.1, 20,50, 1,2, \
20.0,20.0,20.0, \
49.4,49.4,47.6, \
\
9.4, 8.3, 7.5, \
43.1,34.4,28.6)

returns 34.4

eg
table 1
0.02 0.06 0.1
-----
20 20.0 20.0 20.0
50 49.4 49.4 47.6

table 2
0.02 0.06 0.1
-----
20 9.4 8.3 7.5
50 43.1 34.4 28.6
-
|
notes

indexes must be given in ascending order and null values are not
completely handled, any key is null null is returned, unpredictable
if values or indexes contain nulls.

reasonably thoroughly tested on 2 and 3 d tables, and a bit on
1 d tables

n_lookupn()
checks the arguments as far as possible
=====
x_lookupn(argc, argv, xcell, ncells)
double *argv[];
double *xcell;
{
int dimensions;
int* dimension;
int max_required_values;
int max_required_indexes;
int arg_pointer=0;
int value_pointer=0;
double* key;
double* index;
double* value_list;
void* table_md;
double* query;

int i,j;

/*for each set of arguments passed*/
while (ncells-- > 0)
{
/*read the number of dimensions*/
dimensions=(int)argv[0][ncells];

if(dimensions<1 || ISNULL_D(&(argv[0][ncells])))
{
fprintf(stderr,"negative/null dimensions to lookup table\n");
SETNULL_D(xcell[ncells]);
break;
}

if(argc<dimensions+1)
{
fprintf(stderr,"error in xlookupn, too few arguments!\n");
n_lookupn("xlookupn");
SETNULL_D(xcell[ncells]);
break;
}

/*allocate space for dimensions
and put dimensions into it,
for various reasons dimensions

```

```

are put in in reverse*/

dimension=(int*)G_calloc(dimensions,sizeof(int));
for(i=0;i<dimensions;i++)
{
    dimension[dimensions-i-1]=(int)argv[i+1][ncols];
}

/*calculate how many value and
index arguments there should be*/

mes_required_values=(int)dimension[0];
mes_required_indexes=(int)dimension[0];
for(i=1;i<dimensions;i++)
{
    mes_required_values=mes_required_values+dimension[i];
    mes_required_indexes=mes_required_indexes+dimension[i];
}

/* if the actual number of arguments
is less than required set xcell to null*/

if(argc<mes_required_values+mes_required_indexes+dimensions*2+1)
{
    us_lookup("xlookup");
    SETNULL_D(xcell[ncols]);
    free(dimension);
    break;
}

/*allocate space for keys
and put keys into it*/

key=(double*)G_calloc(dimensions,sizeof(double));
for(i=0;i<dimensions;i++)
{
    key[i]=argv[i+1+dimensions][ncols];
}

/*allocate space for indexes
and put index into it*/

arg_pointer=i+2+dimensions;
index=(double*)G_calloc(dimensions,sizeof(double));
for(i=0;i<dimensions;i++)
{
    index[i]=(double*)G_calloc(dimension[dimensions-i-1],sizeof(double));
    for(j=0;j<dimension[dimensions-i-1];j++)
    {
        index[i][j]=argv[arg_pointer][ncols];
        arg_pointer++;
    }
}

/*allocate value list
and put values into it this is
a temporary stage just to simplify
later programming*/

value_list=(double*)G_calloc(mes_required_values,sizeof(double));
for(i=0;i<mes_required_values;i++)
{
    value_list[i]=argv[arg_pointer+1][ncols];
}

/*sort out the keys, the
values are going to be put into an n
dimensional array, the keys are standardised
to the dimensions of the array (linearly
interpolating if the keys don't exactly match
an index value) the query is stored in reverse
order in a double array*/

query=(double*)G_calloc(dimensions,sizeof(double));
for(i=0;i<dimensions;i++)
{
    for(j=0;j<dimension[dimensions-i-1];j++)
    {
        if(j==0 && key[i]<index[i][j])
        {
            query[dimensions-i-1]=0;
            break;
        }
        if(j==dimension[dimensions-i-1]-1 && key[i]>index[i][j])
        {
            query[dimensions-i-1]=dimension[dimensions-i-1]-1;
            break;
        }
        if(key[i]<index[i][j])
        {
            query[dimensions-i-1]=(key[i]-index[i][j-1])/
            (index[i][j]-index[i][j-1])+j-1;
            break;
        }
    }
}

/*allocate space for an nd table, this
is a recursive algorithm*/

```

```

    table_nd=recurs_allocate(dimensions,dimension,0);

/*put values into table*/
    recurs_pack(table_nd,dimensions,dimension,0,value_list,0);

/*query the table*/
    xcell[ncols]=recurs_linear_get(table_nd,dimensions,dimension,0,query);

/*free everything*/
    recurs_free(table_nd,dimensions,dimension,0);
    free(query);
    free(key);
    for(i=0;i<dimensions;i++)
    {
        free(index[i]);
    }
    free(index);
    free(dimension);
    free(value_list);
}

return i;
}

n_lookupn(n,name) char *name;
{
    if (n < 5)
    {
        us_lookupn(name);
        return 0;
    }

/*looks okay return 1*/
    return 1;
}

/*****
n_lookupr()
n-dimensional lookup function with see also:
lookup() a 1 dimensional lookup function
lookupn() n-dimensional lookup function

this function works like lookupn except than the first key given
is a key into the value arguments rather than the first index, and
the corresponding index is returned:

lookup(number of dimensions (n),
length of dimension 1(l1),...,length of dimension n(ln),
key 1,...,key2,
    index(d1)1,...,index(d1)n,...,index(dn)1,...,index(dn)n,
    value(d1)1,...,value(d1)1n,...,value(dn)1,...,value(dn)n)

a 2-d example (basic 2d table)
this table is table 27 from van Keulen and Wolf(1988)
"Modelling of agricultural Production, Weather, Soil and Crops"
PUDOC Wageningen. It is reversed to give an ascending index

lookup(2,8,10, 75,200,\
0.02, 0.06, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50,\
20,50,100,250,500,1000,2500,5000,10000,16000\
20.0, 19.9, 19.8, 19.7, 19.5, 19.3, 19.1, 18.9,\
49.7, 49.1, 48.6, 47.9, 47.3, 46.0, 44.9, 43.8,\
96.8, 91.1, 86.8, 82.4, 78.9, 73.3, 69.0, 65.4,\
165.6,126.6,111.2,100.1, 92.7, 83.0, 76.4, 71.6,\
218.3,147.3,124.0,108.8, 99.3, 87.4, 79.8, 74.2,\
264.7,163.8,134.1,115.6,104.4, 90.8, 83.3, 76.2,\
307.0,178.2,142.7,121.3,108.8, 93.7, 84.5, 78.0,\
332.1,186.7,147.8,124.7,111.3, 95.4, 85.8, 79.0,\
352.6,193.0,151.9,127.6,113.4, 96.8, 86.8, 79.8,\
383.4,197.1,154.1,128.9,114.4, 97.5, 87.3, 80.3)

returns 0.37659
0.37659
200
| 0.02 0.06 0.1 0.15 0.2 | 0.3 0.4 0.5
|-----|
| 20 20.0 19.9 19.8 19.7 19.5 | 19.3 19.1 18.9
| 50 49.7 49.1 48.6 47.9 47.3 | 46.0 44.9 43.8
| 100 96.8 91.1 86.8 82.4 78.9 | 73.3 69.0 65.4
-----> 75
250 165.6 126.6 111.2 100.1 92.7 83.0 76.4 71.6
500 218.3 147.3 124.0 108.8 99.3 87.4 79.8 74.2
1000 264.7 163.8 134.1 115.6 104.4 90.8 83.3 76.2
2000 307.0 178.2 142.7 121.3 108.8 93.7 84.5 78.0
5000 332.1 186.7 147.8 124.7 111.3 95.4 85.8 79.0
10000 352.6 193.0 151.9 127.6 113.4 96.8 86.8 79.8
16000 383.4 197.1 154.1 128.9 114.4 97.5 87.3 80.3

notes
only checked on a 2d table, 1d can be handled with lookup()
nulls not handled except as keys when a null is returned

n_lookupr()

```



```

checks the arguments as far as possible
*****
x_lookupr(argc, argv, xcell, ncols)
    double *argv[];
    double *xcell;
    {
        int dimensions;
        int* dimension;
        int nos_required_values;
        int nos_required_indexes;
        int arg_pointer=0;
        int value_pointer=0;
        double* key;
        double** index;
        double* value_list;
        void* table_md;
        double* query;
        double index_query;
        int i,j;

/*for each set of arguments passed*/

        while (ncols-- > 0)
        {

/*read the number of dimensions*/

            dimensions=(int)argv[0][ncols];

            if(dimensions<1 || ISNULL_D(&(argv[0][ncols])))
            {
                fprintf(stderr,"negative/null dimensions to lookup table\n");
                SETNULL_D(xcell[ncols]);
                break;
            }

            if(argc<dimensions+1)
            {
                fprintf(stderr,"error in xlookup, too few arguments!\n");
                us_lookupr("xlookup");
                SETNULL_D(xcell[ncols]);
                break;
            }

/*allocate space for dimensions
and put dimensions into it,
for various reasons dimensions
are put in in reverse*/

            dimension=(int*)G_calloc(dimensions,sizeof(int));
            for(i=0;i<dimensions;i++)
            {
                dimension[dimensions-i-1]=(int)argv[i+1][ncols];
            }

/*calculate how many value and
index arguments there should be*/

            nos_required_values=(int)dimension[0];
            nos_required_indexes=(int)dimension[0];
            for(i=1;i<dimensions;i++)
            {
                nos_required_values=nos_required_values+dimension[i];
                nos_required_indexes=nos_required_indexes+dimension[i];
            }

/* if the actual number of arguments
is less than required set xcell to null*/

            if(argc<nos_required_values+nos_required_indexes+dimensions*2+1)
            {
                us_lookupr("xlookup");
                SETNULL_D(xcell[ncols]);
                free (dimension);
                break;
            }

/*allocate space for keys
and put keys into it*/

            key=(double*)G_calloc(dimensions,sizeof(double));
            for(i=0;i<dimensions;i++)
            {
                key[i]=argv[i+1+dimensions][ncols];
            }

/*allocate space for indexes
and put index into it*/

            arg_pointer=i*2+dimensions;
            index=(double**)G_calloc(dimensions,sizeof(double));
            for(i=0;i<dimensions;i++)
            {
                index[i]=(double*)G_calloc(dimension[dimensions-i-1],sizeof(double));
                for(j=0;j<dimension[dimensions-i-1];j++)
                {
                    index[i][j]=argv[arg_pointer][ncols];
                    arg_pointer++;
                }
            }

```

```

    }
}

/*allocate value list
and put values into it this is
a temporary stage just to simplify
later programming*/

value_list= double* calloc(nos_required_values,sizeof(double)),
for(i=0; i<nos_required_values; i++)
{
    value_list[i]=argv[arg_pointer+i][ncols]
}

/*sort out the keys, the
values are going to be put into an n
dimensional array, the keys are standardised
to the dimensions of the array (linearly
interpolating if the keys don't exactly match
an index value) the query is stored in reverse
order in a double array*/

query= double* calloc(dimensions,sizeof(double)) ,
for(i=1; i<dimensions; i++)
{
    for(j=0; j<dimension[dimensions-i-1]; j++)
    {
        if(j==0 && key[i]<index[i][j])
        {
            query[dimensions-i-1]=0,
            break
        }
        if(j==dimension[dimensions-i-1]-1 && key[i]>=index[i][j])
        {
            query[dimensions-i-1]=dimension[dimensions-i-1]-1,
            break
        }
        if(key[i]<index[i][j])
        {
            query[dimensions-i-1]=(key[i]-index[i][j-1])/
            (index[i][j]-index[i][j-1])*j-1,
            break
        }
    }
}

/*allocate space for an nd table, this
is a recursive algorithm*/

table_nd=recurse_allocate(dimensions, dimension, 0);

/*put values into table*/

recurse_pack(table_nd, dimensions, dimension, 0, value_list, 0);

/*query the table*/

ncall[ncols]=recurse_reverse_get(table_nd, dimensions, dimension, 0, query, key[0], index[0]);

/*free everything*/

recurse_free(table_nd, dimensions, dimension, 0);
free(query);
free(key);
for(i=0; i<dimensions; i++)
{
    free(index[i])
}
free(index)
free(dimension)
free(value_list)
}
return i;
}

n_lookup(n, name char *name;
{
    if (n < 5)
    {
        n_lookup(name ,
        return 0;
    }
}

/*looks okay return 1*/
return 1;
}

n_lookup(name)
char* name,
{
    printf(stderr, "Is too few arguments\n", name);
    printf(stderr, "usage  %s  %d, %d, ..., %d, %s", name ,
    printf(stderr, "          key1, ..., keyd, %s",
    printf(stderr, "          index11, ..., index1i1, ..., indexdi1, ..., indexdi4\n");
    printf(stderr, "          value1 1-1, ..., value11 12-1d\n");
}

void* recurse_allocate(n, list, pos)
int n, /*number of dimensions*/

```

```

int* list; /*list of dimension lengths*/
int pos; /*position in list*/
{
    int i;
    double** temp;
    int new_pos;

    /*if this is the bottom
    level allocate a double size memory
    and return pointer*/

    if(pos==n)
    {
        return (void*)calloc(sizeof(double));
    }

    /*increment the position in the list*/

    new_pos=pos+1;

    /*allocate an array of pointers
    to double*/

    temp=(double**)calloc(list[pos],sizeof(double));

    /*for each pointer in array
    recursively call this function
    to allocate memory*/

    for(i=0;i<list[pos];i++)
    {
        temp[i]=(double*)recurse_allocate(n,list,new_pos);
    }

    /*return pointer to array of pointers*/

    return (void*)temp;
}

int recurse_pack(pointer,n,list,pos,value,value_pos)
void* pointer; /*pointer to n dimensional table*/
int n; /*number of dimensions (or values in list)*/
int* list; /*list of dimension sizes*/
int pos; /*current position in list*/
double* value; /*list of values to put in table*/
int value_pos; /*current position in value list*/
{
    int new_pos=0;
    int i;
    double** temp;

    /*if this is the bottom level
    put the current value into
    the bit of memory pointed at by pointer
    increment the value position and return it*/

    if(pos==n)
    {
        *((double*)pointer)=value[value_pos];
        value_pos=value_pos+1;
        return value_pos;
    }

    /*point temp a same memory as pointer*/

    temp=(double**)pointer;

    /*increment position in list*/

    new_pos=pos+1;

    /*for each pointer in the
    memory pointed at by pointer recursively
    pack in the values*/

    for(i=0;i<list[pos];i++)
    {
        value_pos=recurse_pack((void*)temp[i],n,list,new_pos,value,value_pos);
    }

    /*return the current value position*/

    return value_pos;
}

recurse_free(pointer,n,list,pos)
void* pointer; /*pointer to n dimensional table*/
int n; /*number of dimensions*/
int* list; /*list of dimension lengths*/
int pos; /*position in list*/
{
    int i;
    double** temp;
    int new_pos;

    /*if this is the bottom level free
    memory*/

    if(pos==n)

```



```

    {
        free(pointer);
        return;
    }

/*increment position in list*/
    new_pos=pos+1;

/*point temp at same memory pointed
at by pointer*/
    temp=(double**)pointer;

/*recursively free each bit of memory
pointed at by the pointers that pointer
points at (sic)*/
    for(i=0;i<list[pos];i++)
    {
        recurs_free((void*)temp[i],n,list,new_pos);
    }

/*now free pointer*/
    free(pointer);
}

double recurs_linear_get(pointer,n,list,pos,query)
void* pointer; /*pointer to a dimension table*/
int n; /*number of dimensions*/
int* list; /*list of dimension lengths*/
int pos; /*current position in length*/
double* query; /*list of query values*/
{
    int lower; int higher;
    int new_pos;
    double** temp;

/*convert query to nearest
higher and lower values*/
    lower=(int)query[pos];
    higher=lower+1;

/*point temp at memory pointed
at by pointer*/
    temp=(double**)pointer;

/*if this is the bottom level
return value pointed at by pointer*/
    if(pos==n)
    {
        return *((double**)pointer);
    }

/*increment new_pos*/
    new_pos=pos+1;

/*if higher integer (or lower integer
but in that case there is a major error)
is higher than highest possible value
(ie the size of the current dimension -1)
recursively return the value pointed at by
the pointer[lower] pointer at by pointer*/
    if(higher>list[pos] || lower>list[pos])
    {
        return recurs_linear_get(temp[list[pos]-1],n,list,new_pos,query);
    }

/*return the linear interpolation
of the values returned from pointer[higher]
and pointer[lower]*/
    return(query[pos]-lower)*
    (recurs_linear_get(temp[higher],n,list,new_pos,query)-
    recurs_linear_get(temp[lower],n,list,new_pos,query)) +
    recurs_linear_get(temp[lower],n,list,new_pos,query);
}

double recurs_reverse_get(pointer,n,list,pos,query,key,index)
void* pointer; /*pointer to a dimension table*/
int n; /*number of dimensions*/
int* list; /*list of dimension lengths*/
int pos; /*current position in length*/
double* query; /*list of query values*/
double key; /*key to search through values*/
double* index; /*index list from which to derive return value*/
{
    int lower; int higher;
    int new_pos;
    double** temp;
    double value,last_value;
    double value_high,value_low;
    int i;

```

```

/*convert query to nearest
higher and lower value*/

lower=(int)query[pos];
higher=lower+1;

/*point temp at memory pointed
at by pointer*/

temp=(double**)pointer;

/*if this is one up from bottom level
search through the list of values pointed
at by pointer and return a linear interpolation
of the corresponding index value,
or return an end index if this key is outside
range of values*/

if(pos==n-1)
{
    if(list[pos]==1)
    {
        return 0;
    }
    last_value=temp[0];
    for(i=1;i<list[pos];i++)
    {
        value=temp[i];
        if((key>last_value && key<=value) || (key<=last_value && key>=value))
        {
            return((key-last_value)/(value-last_value)*(index[i]-index[i-1])+
index[i-1]);
        }
        last_value=value;
    }
    last_value=temp[0];
    value=temp[list[pos]-1];
    if((last_value==value && value==key) ||
    (last_value<=value && value<=key) )
    {
        return index[list[pos]-1];
    }
    if(value>=last_value && last_value==key ||
    value<=last_value && last_value==key )
    {
        return index[0];
    }
}

/*increment new_pos*/

new_pos=pos+1;

/*linearly interpolate between
the two values above and below query*/

if(higher>=list[pos] || lower>=list[pos])
{
    return(recurs_reverse_get(temp[list[pos]-1],n,list,new_pos,query,key,index));
}

value_high=recurs_reverse_get(temp[higher],n,list,new_pos,query,key,index);
value_low=recurs_reverse_get(temp[lower],n,list,new_pos,query,key,index);
return ((value_high-value_low)*(query[pos]-lower/higher-lower))+value_low;
}

```

A.3.3 Normal random number functions

The Box-Muller algorithm for converting a uniformly distributed random number generator to a normalised random number generator was implemented as an *r.mapcalc* function, see Section 6.7.2 which describes how it can be used to generate stochastic realisations.

```

#include "glob.h"
#include <math.h>
/*Adapted by Thomas Charnock
this is an adaptation of H. Shapiro's x_rand code to produce
normal rand numbers, here is his comment*/

/* Author Michael Shapiro, 16 feb 1993
* NOTE
* This depends on the Unix function drand48() which returns a
* double between 0.0 and 1.0
* a hook is provided for seeding the random numbers by the main()
* g_randseed() it does nothing at present
*/

extern double drand48();
double got_normal_random();

g_randseed()
{

```

```

long seedval;

seedval=time(0);
srand48(seedval);
}

x_rand_mb (argc, argv, cell, ncols)
double *argv[];
register double *cell;
register int ncols;
{
register int i;
double mean, stdev;
double x;

if(ISNULL_D(&mean) || ISNULL_D(&stdev))
{
SETNULL_D(&cell[i]);
}

for (i = 0; i < ncols; i++)
{
x = get_normal_random();
mean = argv[0][i];
stdev = argv[1][i];
if (mean==0 && stdev==1)
{
cell[i]=x;
}
else
{
cell[i] = (x*stdev)+mean;
}
}
}

/*
The Box-Muller algorithm for getting a normal random(0,1)
number from a random number between 0 and 1
taken from Numerical Recipes Press et al in FORTRAN
translated to C by Thomas Charnock*/
double get_normal_random()
{
static double last_value;
static int last_flag=0;

double v1,v2,r,fac,temp;
long int temp2;

if(last_flag)
{
last_flag=0;
return last_value;
}
else
{
do
{
v1=(2.0*drand48()-1);
v2=(2.0*drand48()-1);
r=(v1*v1)+(v2*v2);
}
while(r>1);
fac=sqrt(-2.0*log(r)/r);
last_value=v1*fac;
last_flag=1;
return(v2*fac);
}
}

a_rand_mb (n,name) char *name;
{
if (n==2)
{
return 1;
}

fprintf (stderr, "%s - ", name);
if (n==0)
{
fprintf (stderr, "no arguments ");
}
else if (n<2)
{
fprintf (stderr, "less than two arguments ");
}
else
{
fprintf (stderr, "more than two arguments ");
}

fprintf (stderr, "specified. usage: %s(mean,stdev)\n", name);
return 0;
}

```


A.3.4 Simpson's rule function

Simpson's rules are numerical methods for integrating functions, see for example Chapra and Canale (1985). In Section 4.4 the total soil moisture loss due to drawdown is calculated by using the area under the function given by Hedges (1989). Two new operations were added to *r.mapcalc*. The first simply applies Simpson's $\frac{1}{3}$ rule specifically to Equation 2.6 the other was a general operation to apply Simpson's $\frac{1}{3}$ rule to any equation.

The first specific operation was called *soil*, and implementation involved adding functions as has been explained above, it is documented at the end of this section. The second and more general operation involved changing *r.mapcalc* at a more fundamental level.

The first task that *r.mapcalc* performs is to pass the expression given by the user to a program which converts it to *polish* notation. Polish notation differs from standard mathematical notation in that the formula is evaluated from left to right, as each operator is reached it is applied to the preceding one or two arguments; thus the formula:

$$A + (B \times C)$$

is converted to:

$$BC \times A +$$

This is a very convenient form when formulae are being evaluated by a computer. *r.mapcalc* attaches a label to each token in the expression to indicate what it is (e.g. a raster name, a binary operator, an integer or a variable) *r.mapcalc* then evaluates the expression for each pixel, in the current region.

r.mapcalc assumes that all tokens labelled as strings are references to rasters, and if a matching raster is not found then an error is indicated. It was necessary to change this so that a string could be passed to a function. The string would be the equation that was to be evaluated by Simpson's rule.

A prototype version of this was developed the symbol \sim was used to tell the *r.mapcalc* parser to ignore the following string. The code for this is not included as the modification required were scattered through out the existing *r.mapcalc* code. In the prototype the second level of equation parsing was done by starting another *r.mapcalc* process this is clearly a very inefficient way of handling this, but valid in a prototype.

The simpsons $\frac{1}{3}$ function has the syntax:

```
simpson3(~"expression",~"variable",low_limit,upper_limit,nos_segments,~"raster")
```

where:

expression is the expression to be evaluated;

variable is the variable which the differentiation is in respect to;

low_limit & upper_limit are the limits of the integration;

raster is a temporary raster name required in this prototype because the process spawns another *r.mapcalc* process to evaluate the expression.

The *soil* function code follows:

```
#include "glob.h"
#include "math.h"

#define DEFAULT_NOS_SEGS 10;

x_soil (argc, argv, xcell, ncols)
double *argv[];
double *xcell;
{
    double hc,hl,a,b,theta_s,theta_r;
    register int n,i;
    int nos_segs;
```

```

double* fx;
double range;
double interval;
double h,sum_of_odds,sum_of_evens;
int doit;

n = ncols;
while(n-- > 0)
{
    doit=1;

    /*sort out arguments*/

    hc=argv[0][n];
    hl=argv[1][n];
    a=argv[2][n];
    b=argv[3][n];

    if(argv[0]==argv[1])
    {
        xcell[n]=0;
        doit=0;
    }

    theta_s=argv[4][n];
    theta_r=argv[5][n];

    if(theta_s==theta_r)
    {
        SETNULL_D(&xcell[n]);
        doit=0;
    }

    if(argc==7)
    {
        nos_segs=(int)argv[6][n];
        if(nos_segs!=2)
        {
            nos_segs++;
        }
        if(nos_segs<2)
        {
            nos_segs=DEFAULT_NOS_SEGS;
        }
    }
    else
    {
        nos_segs=DEFAULT_NOS_SEGS;
    }

    /*is any of the first six arguments
       is null set xcell[n] to null*/

    if (ISNULL_D(&hc) ||
        ISNULL_D(&hl) ||
        ISNULL_D(&a) ||
        ISNULL_D(&b) ||
        ISNULL_D(&theta_s) ||
        ISNULL_D(&theta_r))
    {
        SETNULL_D(&xcell[n]);
        doit=0;
    }

    /*if everything is okay then do it*/

    if(doit)
    {
        floating_point_exception = 0;

        /*allocate some space for intermediate values
           for each segments*/

        fx=(double*)C_malloc(nos_segs+1,sizeof(double));

        /*work out the range and the interval*/

        range=hc-hl;
        interval=range/nos_segs;

        /*set initial values*/

        h=hl;
        sum_of_odds=0;
        sum_of_evens=0;

        /*work out f(x) for each segment boundary*/

        for(i=0;i<nos_segs;i++)
        {
            fx[i]=(theta_s-theta_r)*a/(a*pow(h,b))+theta_r;

            if(i!=nos_segs)
            {
                sum_of_odds=sum_of_odds+fx[i];
            }
        }
    }
}

```

```

        if((i%2) && i!=0 && i!=nos_segs)
        {
            sum_of_odds=sum_of_odds+fx[i];
        }

        h=h+interval;
    }

    /*do simpson's 1/3 rule*/
    xcell[n]=range*(fx[0]+4*sum_of_odds+2*sum_of_evens+fx[nos_segs])/(3*nos_segs);

    /*free space and test for floating
    point exceptions*/

    free(fx);
    if(floating_point_exception)
    {
        SETWILL_D(xcell[n]);
    }
}
}
return i;
}

u_soil(n,name) char *name;
{
    if (n == 7 || n == 6) return i;
    fprintf (stderr, "%s - ",name);

    u_soil();

    return 0;
}

u_soil()
{
    fprintf(stderr, "\nusage: soil(hu,hl,s,b,theta_s,theta_r[,nos_segs])\n\n");

    fprintf (stderr, "should be 6 or 7 arguments specified\n\n");

    fprintf(stderr, "\thu - is upper bound (positive, L)\n");
    fprintf(stderr, "\thl - is lower bound (positive, L)\n");
    fprintf(stderr, "\ts - empirical parameters\n");
    fprintf(stderr, "\ttheta_s -- saturated moisture content (fraction)\n");
    fprintf(stderr, "\ttheta_r -- residual moisture content (fraction)\n");
    fprintf(stderr, "\tnos_segs -- number of segments for Simpsons rule\n");
}

```

A.4 GRASS and PERL

PERL is a language that stands half way between being a full programming language and a shell. Several useful applications were written using it. Whilst developing GRASS applications in PERL, several general subroutines were developed for performing GRASS operations in PERL, these have been brought together in the PERL library *grass.pl*. This library of functions is in ongoing development. Where a function is well used it would be worth reimplementing it as low level C code for speed. This would be an example of the gradual development life-cycle described in Section 8.4 and shown in Figure 8.7.

A.4.1 The *grass.pl* library

Some of these are just calls to GRASS commands, some write and read directly from GRASS files, some use specially written C programs and some perform initialisation and parsing tasks. The library is continually being added to as new commands are required. Currently the subroutines are:

- *initiate_new_ascii_raster* begins a new file which can be converted to a true GRASS raster;
- *write_to_ascii_raster* writes values into ascii raster;
- *write_ascii_raster_to_database* converts ascii raster to true grass raster;

- *get_temp_file* creates a temporary file name;
- *write_site_file* writes sites to a site file;
- *read_site_file* reads sites from a site file;
- *sample_raster* gets the values of a raster from a point;
- *get_easting* converts a column number into an easting value;
- *get_northing* converts a row number into a northing value;
- *set_region* sets the current GRASS region;
- *get_raster_region* gets the region of a GRASS raster;
- *get_current_region* gets the current GRASS region;
- *find_file* searches GRASS database, or given mapset for a file of specified type;
- *initiate* initiates the PERL program to work with GRASS, also sets up command line options and flags;
- *parse_options* a commandline parser function that has similar behaviour to the GRASS parser (Appendix A.1.1);
- *usage* writes the correct syntax of the program (e.g. command line flags and options) to the screen.

Many programs used the library of functions a good example is the *s.rms* program.

A.4.2 The *s.rms* program

s.rms is an *ad hoc* script designed to calculate the root mean square error, between a raster and a site file. The raster might be a modelled water table surface and the site file might contain real observations (see the calibration script in Section 5.8.2). The script can also generate a site file containing error at each site location. This script is very inefficient for large site files (as it requires the raster to be opened and closed at each location), if it was frequently used then it would be sensible to re-implement it in C. This is another example of the gradual development life-cycle described in Section 8.4 and shown in Figure 8.7.

```
#!/usr/local/bin/perl

#find the root-mean-square error between a site file and a raster
#optionally creates a new site file with absolute errors

do 'grass.pl';
do 'useful.pl';

$program_name="s.rms";
$option_list=('input','output','map');
$option_desc_list=('input site file',
'output site file',
'raster map to sample');
$default_list=($null_value,$null_value,$null_value);
$flag_list=('r','m','a');
$flag_desc_list=('just sites in current region',
'mask out areas with MASK',
'make errors absolute ie +ve');
$flag_default_list=('0','0','1');

#check that this PERL script
#is running through grass

#initiate;

#parse the command line options
#parse_options();
```

```

#make sure an input site file
#and a raster map are specified

if($opt{"input"} eq $null_value ||
    $opt{"map"} eq $null_value)
{
    printf(STDERR "Error: must set input site file and map\n");
    exit 1;
}

#read site file into site structure
if(!($insites=read_site_file($opt{"input"})))
{
    printf(STDERR "Error: reading site file <$opt{'input'}>\n");
    exit 1;
}

#extract the site list from site structure
@in_site_list=@{$insites{"sites"}};

#if mask flag set check if mask exists
if($flag{"m"})
{
    if(!find_file("MASK","cell",$env_list{"MAPSET"}))
    {
        printf(STDERR "no MASK found\n");
        $flag{"m"}=0;
    }
}

#get current region and initiate
#count and total_sq_diff

$reg=$get_current_region();
$count=0;
$total_sq_diff=0;

#for each site:
foreach $site (@in_site_list)
{
    #get east north and description
    ($east,$north,$desc)=split(/\/,$site);

    #if description is a valid number
    #sample the raster at $east and $north

    if($flag{"m"})
    {
        $mask_value=$sample_raster("MASK",$east,$north);
    }

    if($is_number($desc) &&
        !($flag{"r"} &&
        ($east<$reg{"west"} || $east>$reg{"east"} ||
        $north<$reg{"south"} || $north>$reg{"north"})) &&
        !($flag{"m"} && ($mask_value eq 'null' || $mask_value == 0)))
    {
        $map_value=$sample_raster($opt{"map"},$east,$north);

        #if sample raster return undef
        #than an invalid raster has been
        #given so exit

        if(!defined($map_value))
        {
            printf(STDERR "Error: map <$s> not found\n",$opt{"map"});
            exit 1;
        }

        #if map_value is not null
        #square the difference between map_value
        #and site description and add to running total
        #if an output site is specified
        #then find the root of the square different
        #and add to the end of the out_list_list
        #increment the count

        if($map_value ne "null")
        {
            $diff=$map_value-$desc;
            $sq_diff=$diff*$diff;
            $total_sq_diff=$total_sq_diff+$sq_diff;
            if($opt{"output"} ne $null_value)
            {
                $root_sq_diff=sqrt($sq_diff);
                if($flag{"a"})
                {
                    $out_site_list[$count]="$east|$north|$root_sq_diff";
                }
                else
                {
                    $out_site_list[$count]="$east|$north|$diff";
                }
            }
        }
    }
}

```

```

    }
    }
    $count++;
  }
}

#workout and print rms

if($count>0)
{
  $root_mean_sq_diff=sqrt($total_sq_diff/$count);
  printf(STDOUT"%g\n",$root_mean_sq_diff);

  #if an output site file is specified
  #then write it

  if($opt{"output"} ne $null_value)
  {
    $outsites{"desc"}="$rms_$opt{input}_$opt{map}":$root_mean_sq_diff;
    $outsites{"sites"}=$out_site_list;
    $write_site_list($opt{"output"},\%outsites);
  }
}
else
{
  printf(STDERR"Error: no sites with numbers found\n");
}
exit 0;

```

A.4.3 The *r.swms.region* program

A controlling program for *r.swms* was developed to run the program at each grid square across a region. The program is called *r.swms.region* and is documented in Appendix C.2. an interesting feature of the program is that it can spawn processes on several machine thus running the model in parallel and reducing the time take to run. The program was written in PERL and used the PERL function library documented above (Section A.4.1).

```

#!/usr/local/bin/perl -w

#find the root-mean-square error between a site file and a raster
#optionally creates a new site file with absolute errors

use POSIX "sys_wait_h";
do 'grass.pl';
do 'useful.pl';

#get this machine name

chop($hostname = `hostname`);

$program_name="r.swms.region";
$option_list=('machines','template','dataset','error','sleep');
$option_desc_list= "hosts to perform swms modelling",
'template raster to mark areas to be modelled',
'dataset: construct with r.swms',
'file to send message generated by SWMS_2d',
'period to sleep between starting processes and checking they are finished';
$default_list=($hostname,$null_value,'noname',$null_value,10);
$flag_list=('q');
$flag_desc_list=('quiet: suppress dialogue');
$flag_default_list=(0);

#check that this PERL script
#is running through grass

initiate;

#parse the command line options

parse_options();

#check that the dataset exists
#using GRASS command g.findfile
#if it doesn't then exit

$dataset_mapset=find_file($opt{'dataset'},"swms_datasets");
if(!$dataset_mapset)
{
  printf(STDERR"$opt{'dataset'} is not a valid dataset\n");
  $usage;
  exit;
}

#check a machine list is given

if($opt{'machines'} eq $null_value)
{
  printf(STDERR"give a list of machines\n");
  $usage;
}

```



```

    exit;
}

#check template exists

$use_template=0;
$template_mapset=find_file($opt{'template'},"cell");
if($template_mapset)
{
    $use_template=1;
}

#get current region and if
#s template is used get template region

$current_region=$get_current_region;

if(!$use_template)
{
    $model_region=$current_region;
}
else
{
    $model_region=$get_raster_region($opt{'template'},$template_mapset);
    $set_region($model_region);
}

#sort out which machines to run sums on

if($opt{'machines'} ne $null_value)
{
    $machines=split(/,/,$opt{'machines'});
}
else
{
    printf(STDERR"Error: specify a list of machines to run on\n");
}

#redirect stdout, sums produces
#s lot of text!

open (SAVEOUT, ">&STDOUT");
close (STDOUT);
if($opt{'error'} ne $null_value)
{
    open (STDOUT,">$opt{'error'}");
}

#for each pixel of template run
#r.sums to generate input files
#then run sums

$spath=$env_list(GISDBASE)/$env_list(LOCATION_NAME)/$env_list(MAPSET)/sums_run/$opt{'dataset'};

#remove old sums output files

system("rm -r $spath\n");

if(!$flag{'q'})
{
    printf(STDERR"Writing input files....\n");
}

for($i=0,$i<$model_region{'rows'};$i++)
{
    $northing=$get_northing($model_region,$i+0.5);
    for($j=0,$j<$model_region{'cols'};$j++)
    {
        $easting=$get_easting($model_region,$j+0.5);
        system("r.sums dataset=$opt{'dataset'} x=$easting y=$northing -q");
    }
}

#call sums_2d to do for every pixel

if(!$flag{'q'})
{
    printf(STDERR"Calling sums_2D \n");
}

$ntoo_left=1;
$prcoesses_left=1;
while($prcoesses_left||$ntoo_left)
{
    if($ntoo_left)
    {
        $ntoo_left=$get_prcoesses_running();
    }
    sleep(10);
    $prcoesses_left=$get_prcoesses();
}

close(STDOUT);
open (STDOUT,">&SAVEOUT");

exit 0;

sub get_prcoesses

```

```

{
my ($exit_status, $machine, $found);

while (($pid = waitpid(-1, &WNOHANG)) > 0)
{
$exit_status=$?;

$found=0;
foreach $machine (@machines)
{
if(defined($machine_pid($machine)) && $pid==$machine_pid($machine))
{
if(!flag('q'))
{
printf(STDERR"process <$pid> on $machine finished\n");
}
undef($machine_pid($machine));
$found=1;
}
}
if(!found && !flag('q'))
{
printf(STDERR"pid <$pid> not found\n");
}
}

foreach $machine (@machines)
{
if(defined($machine_pid($machine)))
{
return 1;
}
}
return 0;
}

sub set_processes_running
{
my ($machine,$site);

foreach $machine (@machines)
{
if(!defined($machine_pid($machine)))
{
$site=$get_next_site();
if(!defined($site))
{
return 0;
}
$run_sums_2d($site,$machine);
}
}
return 1;
}

sub run_sums_2d
{
my($coords,$pid);

$pid=fork;

if($pid)
{
$machine_pid($_[1])=$pid;
return;
}
else
{
@coords=split(/,/,$_[0]);

if(!flag('q'))
{
printf(STDERR"processing site $_[0] on $_[1]\n");
}
printf(STDERR"rah $_[1] \n~cd $path/$opt{'dataset'}$coords[0]\$_$coords[1] ; rm SVMS_2D.OUT/* ; sums\n");
exit;
}
}

sub get_next_site
{
my( $count );

undef($site);

if (!defined($row))
{
$row=0;
}

if(!defined($col))
{
$col=0;
}

$count=0;

```

```

for($row=$row;$row<$model_region('rows');$row++)
{
    $northing=$get_northing($model_region,$row+0.5);
    for($col=$col;$col<$model_region('cols');$col++)
    {
        if($count==1) { return $site; }
        $easting=$get_easting($model_region,$col+0.5);
        $site="$easting,$northing";
        $count++;
    }
    $col=0;
}
return ($site);
}

```

A.5 RDB

RDB is a set of relational database operators implemented in PERL, see Section A.4. It has been placed in the public domain by its developers (Hobbs, 1993). It has been used to manage the large monitoring borehole database, which was supplied by NRA. The data was stored in two *RDB* files *wells* and *obs* which correspond to the two relation tables in Figure A.4.

A.5.1 Using RDB

RDB follows the relational database paradigm (Avison, 1992).

The principle operators are:

- row — selects rows from an *RDB* table using logic comparisons;
- column — selects columns from an *RDB* table referenced by column number, (*-v* flags selects all columns except those named);
- sorttbl — sorts an *RDB* table on the field name given;
- jointbl — joins two *RDB* tables using the key field name given;
- compute — applies a file of PERL code to an *RDB* table to calculate new field values.

The *RDB* operators are linked together using the UNIX pipe and redirection facilities (Section 3.4.3). For example:

```

row < obs no eq 2306 and year ge 1989 and year le 1990 | \
sorttbl year month day | column -v -a mjd 6N | \
compute -fjulian | \
column mjd wt | headchg -del > wt.dat

```

This query reads the database file *obs* and selects those rows corresponding to observations from borehole '2306' in 1989 and 1990. The results is sorted by year, month and day. An extra blank field "mjd" is added. The "modified julian day" is calculated using a *compute* script *julian*, and put in the blank field "mjd". All fields are removed except watertable 'wt' and 'mjd'. Finally the header is stripped and the output is written to a file "wt.dat" in a form usable by some other facility such as gnuplot.

The *compute* facility of *RDB* is very useful. It allows the full programming capabilities of PERL to be easily applied to the data. A number of *compute* scripts were used during the course of the project.

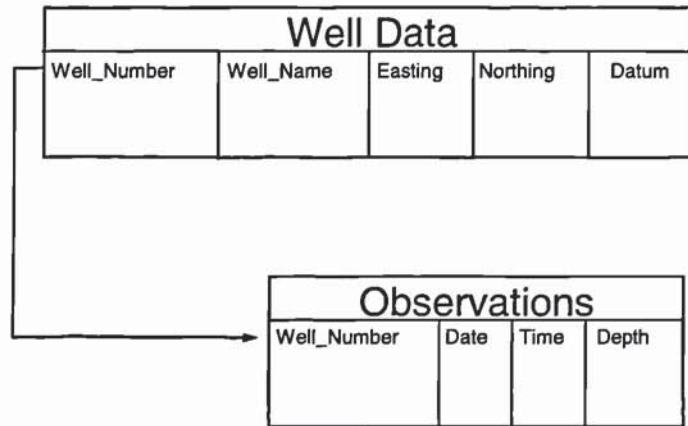


Figure A.4: RDB database structure.

A.5.2 drawdown, julian and time scripts

The *drawdown* script is used by the *compute* operator to work out the relative drawdown of a time sequence of depth to watertable readings, the first reading is assumed to be zero drawdown.

```
if( ! defined( $first_depth ) )
{
    $first_depth = depth ;
}
drawdown = $first_depth - depth ;
```

The *julian* compute script converts *day*, *month* and *year* fields to Julian days. It expects an empty field *mjd* in which to put the calculated Julian day.

```
$compute query for rdb
$given modified julian days
$expects columns: year,month,day and a column mjd to fill
$e.g row <obs no eq 2306 | sorttbl year month day | \
$ column -v -a mjd 6N | compute -fjulian
$formula translated the from FORTRAN code of the Aston University Satellite
$Orbit Group, written by Chris Murphy
use integer;
$stamp1 = ( month - 14 ) /12;
$stamp2 = day - 2432076 ;
$stamp2 = $stamp2 + (1461 * ( year + 4800 + $stamp1)) / 4;
$stamp2 = $stamp2 + (367 * ( month - 2 - ( $stamp1 * 12 )))/12;
$stamp2 = $stamp2 - (3 * (( year + 4900 + $stamp1)/100)) /4;
mjd = $stamp2
```

The *Time* script converts the field 'time' given as a four figure 24hour clock into a fraction of a day. Designed to be used with the *Julian* script.

```
$convert time given in 24hr clock eg 0934 to fraction of day
$hour = substr( time , 0 , 2 );
$minute = substr( time , 2 , 2 );
$minutes = ( $hour * 60 ) + $minute;
time_frac = mjd + ( $ minutes / 1440 );
```

The hydrographs of Figure 5.58 and 5.59 were generated using these to scripts. The syntax

used was:

```
row < obs no eq 2093 and year eq 1984 and month ge 6 and month le 8 | \
sorttbl month day time | \
column -v -a drawdown | compute -fdrawdown | \
column -v -a mjd | compute -fjulian | \
column -v -a time_frac | compute -fTime | \
column time_frac drawdown | headchg -del > drawdown.dat
```

A.5.3 Watertable extraction scripts

Three scripts were written during the course of the project to extract data on watertable heights and depths for various purposes, these were:

- `av_depth`;
- `av_wt`;
- `av_wt_per_month`.

These extract data and write them into a new file.

The `av_depth` compute script extracts depths for wells from an *RDB* table and writes the average to a file in GRASS site format. This script used to generate the site files used to interpolate watertable depth surfaces in Section 5.8.2

```
#compute average depth for a period
#assumes column no (well number) east north and depth are present
#should be sorted on well number e.g.
#row < joined north le 331350 and north ge 324900 and east ge 359100 \
#and east le 365600 and year eq 1984 and month eq 3 and day eq 15 | \
#sorttbl no month day | compute -fav_depth | column no east north depth
#printf averages to file "av_depth.dat" in site file format

if( !defined( $well_number ) )
{
  open(OUTPUT,"av_depth.dat") ;
}

if( !defined($well_number) || no != $well_number ||
    $easting != east || $northing != north )
{
  $noes = 1;
  $fpos = tell(OUTPUT);
  $cum_depth = depth ;
  $av_depth = depth ;
  $well_number = no ;
  $easting = east ;
  $northing = north ;
  seek(OUTPUT,$fpos,0);
  printf(OUTPUT"$easting|$northing|$av_depth\n");
}
else
{
  $noes++;
  $cum_depth = $cum_depth + depth ;
  $av_depth = $cum_depth/$noes;
  seek(OUTPUT,$fpos,0);
  printf(OUTPUT"$easting|$northing|$av_depth\n");
}
```

The `av_wt` compute script extracts watertable heights from an *RDB* table, and writes the average for each well to a file in GRASS site format. This script was used to generate the site files used to interpolate watertable surfaces in Section 5.5.2.

```

#compute average wt for a period
#assumes column no (well number) east north and wt are present
#should be sorted on well number and selected for a period
#row < joined north le 331350 and north ge 324800 and east ge 359100 \
#and east le 365600 and year eq 1984 and month eq 3 and day eq 15 | \
#sorttbl no month day | compute -fav_wt | column no east north depth
#printf averages to file "av_wt.dat" in site file format

if( !defined( $well_number ))
{
  open(OUTPUT,">av_wt.dat" ) ;
}

if( !defined($well_number) || no != $well_number &&
    $easting != east && $northing != north )
{
  $nos = 1;
  $fpos = tell(OUTPUT);
  $cum_wt = wt ;
  $av_wt = wt ;
  $well_number = no ;
  $easting = east ;
  $northing = north ;
  seek(OUTPUT,$fpos,0);
  printf(OUTPUT"$easting|$northing|$av_wt\n");
}
else
{
  $nos++;
  $cum_wt = $cum_wt + wt ;
  $av_wt = $cum_wt/$nos;
  seek(OUTPUT,$fpos,0);
  printf(OUTPUT"$easting|$northing|$av_wt\n");
}

```

The compute script *av_wt.per.month* extracts average watertable and depth for each month in a year and writes them to a tab delineated ASCII text file. The script also samples rasters elevation, slope, landscape position index and distance from rivers. This information was used for the statistical investigation in Section 5.6


```

$compute average wt for a each month in a year
$assumes columns no (well number) east north month and wt are present
$should be sorted on no month and selected for a year
$row < joined year eq 1990 and north le 341000 and north ge 312000 and \
$east le 377000 and 348000 and wt nmat /NULL/ | sorttbl no east north month | \
$column no month east north wt depth | compute -faverage_wt_per_month2
$printf averages to file "av_wt.dat" in \tab delineated format

if( !defined( $well_number ) && !defined( $month_number ) )
{
  open(OUTPUT,">av_wt.dat" );
}

if( ( !defined($well_number) || no != $well_number ) &&
    ( $easting != east && $northing ne north ) )
{
  $well_number = no ;
  $easting = east ;
  $northing = north ;
  $dem="r.sample map=dem coord=$easting,$northing'; chomp($dem);
  $lpos="r.sample map=lpos coord=$easting,$northing'; chomp($lpos);
  $dist="r.sample map=stream.dist coord=$easting,$northing'; chomp($dist);
  $slope="r.sample map=slope coord=$easting,$northing'; chomp($slope);

  for($i=0;$i<12;$i++)
  {
    $av_wt[$i]=null;
    $cum_wt[$i]=null;
    $av_dep[$i]=null;
    $cum_dep[$i]=null;
    $nos[$i]=0;
  }
  $fpos = tell(OUTPUT);

  $month_number = month ;
  $cum_wt[$month_number] = wt ;
  $av_wt[$month_number] = wt ;
  $cum_dep[$month_number] = depth ;
  $av_dep[$month_number] = depth ;
  $nos[$month_number] = 1 ;

  seek(OUTPUT,$fpos,0);

  printf(OUTPUT"$easting\t$northing\t$well_number");
  for($i=1;$i<12;$i++)
  {
    printf(OUTPUT"\t$av_wt[$i]\t$av_dep[$i]")
  }
  printf(OUTPUT"\t$dem\t$lpos\t$dist\t$slope \n");
}
else
{
  if($month_number != month )
  {
    $month_number = month ;
    $cum_wt[$month_number] = wt ;
    $av_wt[$month_number] = wt ;
    $cum_dep[$month_number] = depth ;
    $av_dep[$month_number] = depth ;
    $nos[$month_number] = 1;
    seek OUTPUT,$fpos,0);
    printf(OUTPUT"$easting\t$northing\t$well_number");
    for($i=1;$i<12;$i++)
    {
      printf(OUTPUT"\t$av_wt[$i]\t$av_dep[$i]")
    }
    printf(OUTPUT"\t$dem\t$lpos\t$dist\t$slope \n");
  }
  else
  {
    $nos[$month_number]++;
    $cum_wt[$month_number] = $cum_wt[$month_number] + wt ;
    $av_wt[$month_number] = $cum_wt[$month_number]/$nos[$month_number];
    $cum_dep[$month_number] = $cum_dep[$month_number] + depth ;
    $av_dep[$month_number] = $cum_dep[$month_number]/$nos[$month_number];

    seek(OUTPUT,$fpos,0);
    printf OUTPUT"$easting\t$northing\t$well_number");
    for $i=1;$i<12;$i++
    {
      printf(OUTPUT"\t$av_wt[$i]\t$av_dep[$i]")
    }
    printf(OUTPUT"\t$dem\t$lpos\t$dist\t$slope \n");
  }
}

```

Appendix B

Modflow-GRASS link documentation

The Modflow to GRASS link was described in Section 5.4.3 and illustrated in Figure 3.12. Here the programs that form that link are documented. The link comprises programs:

- *mod.edit* — edits the Modflow dataset;
- *mod.write* — writes the Modflow dataset to Modflow input files;
- *mod.run* — a script calls Modflow;
- *mod.del* — deletes a Modflow dataset;
- *mod.rename* — renames a Modflow dataset;
- *mod.copy* — copy a Modflow dataset;
- *mod.d.grid* — displays an irregular grid in the current GRASS monitor;
- *mod.xtrct* — extracts data from the Modflow output files and writes them into the GRASS database;
- *mod.cre* — non-interactively creates an empty Modflow dataset;
- *mod.region* — changes current GRASS region to a Modflow dataset region;
- *mod.rem* — performs a bulk remove of rasters;
- *mod.ts* — extracts values from a family of rasters from a point to form a time series.

B.1 *mod.edit*

mod.edit is the main program for structuring the data within the GRASS database into a dataset. As discussed in Section 3.5.1, the strategy followed has been one of minimum effort. The interface here was constructed using the GRASS “visual ask” (vask) library. This is a very simply library of functions for constructing “form” type of interfaces. The program is started by typing:

```
GRASS 4.1 > mod.edit
```

```

Enter model parameters:
      north edge 331350_____

      |-----|
      |         |
west edge |         | east edge
359100___|         |365600_____
      |         |
      |-----|

      south edge 324800_____
      rows      131          ew res  50_____
      columns   130          ns res  50_____

      layers    1_____ stress periods 1_____

time type:(0-undefined, 1-secs, 2-mins, 3-hours, 4-days, 5-years) 4_

use a grid file y/n? n
      AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

```

Figure B.1: Region definition screen.

or

GRASS 4.1 > mod.edit dataset_name

If the dataset does not exist already then the user is presented with a form in which the basic model dimensions can be defined, Figure B.1. Default are values taken from the current GRASS region (i.e. north, south, east and west edges, and north-south resolution and east-west resolution).

If the an irregular grid is chosen the user is presented with forms to specify the irregular grid (Figure B.2, B.3 and B.4). This part of the interface is still very crude. The first form specifies the number of rows or columns, the next two forms that may cover several pages allow the position of the row and column divides to be specified. The divides at the edges of the region are of course fixed. This is a case where a graphical interface would have made things considerably easier. The command *mod.d.grid* displays on the current GRASS monitor the irregular grid if there is one associated with a dataset.

The next form is the main form (Figure B.5). From here all the forms for defining the dataset can be accessed. Options are selected by typing in the number indicated in the first column. The second column describes the options. An 'X' in the third column indicates an option that must be completed in order to run Modflow. The forth column is the status, this can be either; *not done* — i.e. no data has been added to this option; *complete* — i.e. data has been added and is up to date; *out of date* — i.e. model parameters have been changed since the data in this option was added; and *unsupported* — i.e. this option is not yet supported by the Modflow-GRASS link.

B.1.1 Basic option

Option 1 brings up the forms for specifying the basic input (corresponding to the basic module in Modflow, McDonald and Harbaugh, 1988). Many of the options have a similar form structure. The first form spans two pages, on the first page the user is asked to give a list of rasters which hold the information. In the first form (Figure B.6 and Figure B.7) the user is requested to provide the information to allocate elements to one of three types; *inactive*) i.e. no-flow, *active*

number of columns in grid 217___
number of rows in grid 258___

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

Figure B.2: Grid specification screen 1.

Grid Columns:

352475
352575___
352675___
352775___
352875___
352975___
353075___
353175___
353275___
353375___

10___

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

Figure B.3: Grid specification screen 2.

Grid Rows:

336575	
336475	----
336375	----
336275	----
336175	----
336075	----
335975	----
335875	----
335775	----
335675	----

10----

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

Figure B.4: Grid specification screen 3.

Select a file to edit/create:

	non-optional		status
0	basic input.....	X	complete
1	block centre flow.....	X	not done
2	well.....		not done
3	drains.....		unsupported
4	rivers.....		not done
5	evapotranspiration.....		not done
6	transient leakage.....		unsupported
7	general head boundary.....		unsupported
8	recharge.....		not done
9	strongly implicit procedure.....	X	not done
10	unused.....		unsupported
11	slice successive overrelaxation.....		unsupported
12	quit		
13	to edit model parameters		

X- must be completed
enter a number: 0_

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

Figure B.5: Main option selection screen.

Enter element codes for each layer:				
layer number	one head code only y/n?	if 'y' enter: -1,0 or 1	if 'n' enter raster name for element codes for each layer	valid
1	y	0_-----	-----	y
2	y	0_-----	-----	y
3	y	0_-----	-----	y
4	y	0_-----	-----	y
5	y	0_-----	-----	y
6	y	0_-----	-----	y

-1 -constant 0 -inactive 1 -variable

page 1/2

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

Figure B.6: Element code specification screen.

or *constant*, i.e. fixed head. The first column is the layer number, in the second column the user puts a 'y' to indicate whether a single number or a raster will be provided, i.e. the whole layer can be allocated to one class or a raster can define which class each individual element in. In the third column the user enters the number or alternatively the user gives a raster name in the fourth column. The fifth column indicates whether the data entered is valid or not, when the form is complete the user presses the *escape* key to submit it, if the data is not valid the form is returned with an 'N' next to the invalid data.

The second page allows the users to specify the starting hydraulic heads for each layer and is structured exactly the same as the first page. If there are more than 10 layers the form spans several screens, the user can move up and down the layers by typing the number in the field at the bottom of the page, numbers higher than the number indicates the user has finished filling the form.

The second form (Figure B.8) in the basic option is where the user specifies the lengths of the stress-periods and the time-steps. The first column is the period number (again if there are more than ten periods the form will span more than one screen). In the second column the user specifies the length of the stress period, in the third column the number of time-steps within the stress period is set. The time-step multiplier is specified in the fourth column (the time-step multiplier controls how the stress period is divided into time-steps, see McDonald and Harbaugh, 1988). The fifth column indicates whether the data is valid.

B.1.2 Block centred flow option

The second option is the *block centred flow* (BCF) option. The first screen (Figure B.9) simply allows the user to indicate whether the model is steady state or transient. The choice here governs how the succeeding forms are structured.

The next two forms are presented repetitively for each layer in the model. Figure B.10 shows the first of the two forms, here the layer is assigned to one of four layer types, The anisotropy of the layer is specified (McDonald and Harbaugh, 1988) and, if the model is transient, the primary storage is specified. Again the storage can be a number or a raster.

The second form of the pair allows the other hydrological parameters of the layer to be assigned.

Enter element starting head values for each layer:

layer number	one head value only y/n?	if 'y' enter: a value	if 'n' enter raster name for hydraulic heads for each layer	valid
1	y	0.0000000000	-----	y
2	y	0.0000000000	-----	y
3	y	0.0000000000	-----	y
4	y	0.0000000000	-----	y
5	y	0.0000000000	-----	y
6	y	0.0000000000	-----	y

next layer to edit: 7_

number of layers: 6 page 2/2

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

Figure B.7: Element head specification screen.

Enter the details of each stress period:

period number	length of period	number of time steps	time step multiplier	valid
1	1_	1_	1_	Y
2	1_	1_	1_	Y
3	1_	1_	1_	Y
4	1_	1_	1_	Y
5	1_	1_	1_	Y
6	1_	1_	1_	Y
7	1_	1_	1_	Y
8	1_	1_	1_	Y
9	1_	1_	1_	Y
10	1_	1_	1_	Y

next period to edit 11_

number of periods 12

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

Figure B.8: Stress period specification screen.

```

Enter whether situation is steady state:

is the simulation steady state, y/n? y

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.9: Steady state check screen.

```

Enter initial details for layer:1

layer type: 2

anisotropy: 1-----

one value for      if 'y' what   if 'n' enter raster to      valid
primary storage of  is value      read storage values from
all elements y/n?

y      0.0000000000 ----- y

types are:          for this type 1's storage should be:
0  confined          confined storage coefficient (Sc)
1  unconfined         specific yield (Sy)
2  confined/unconfined confined storage coefficient
3  confined/unconfined confined storage coefficient  page1/2

Sc$=$vol water released from unit vol with unit drop in head
SY$=$vol water released from unit area of aquifer with unit drop in head

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.10: Storage code specification screen.

```

Enter further details for layer:1

one transmissivity if 'y' what if 'n' enter raster to valid
for all rows y/n? is value? read values from
y 0.0000000000 ----- y

one vertical if 'y' what if 'n' enter raster to valid
conductivity 2/n? is value? read values from
y 0.0000000000 ----- y

one secondary if 'y' what if 'n' enter raster to valid
storage y/n? is storage read values from
y 0.0000000000 ----- y

one top level for if 'y' what if 'n' enter raster to valid
whole layer y/n? is height read heights from
y 0.0000000000 ----- y

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.11: Layer properties specification screen.

This form varies depending on whether the model is transient, what layer type the layer is assigned to and the position of the layer (the top and bottom layers have slightly different requirements). Again parameters can be specified as single numbers or parameters.

B.1.3 Well option

The well option is slightly different to the others as it relies in the GRASS site structure. In the first screen (Figure B.12) the user specifies the site file from which to read the spatial data.

Three forms are repetively presented to the user for each site in the site file. In the first (Figure B.13) the information from the site file is presented to the user and the user indicates whether to include the site in the model.

In the second screen (Figure B.14) the user specifies the flow rates from the well for each stress period, and negative value indicates that water is being pumped into the well. If there are more than ten stress periods then the form will span more than one screen.

The third form (Figure B.15) the user divides the flow rates amongst the layers of the models, these numbers should add up to 1. Again if there are more than ten layers the form will span more than one screen.

B.1.4 River option

Unlike the other options the river option relies on the GRASS vector structure. In the first form (Figure B.16) the user specifies which vector to use for the river data. The user also gives a single figure or raster to describe the height of the river and to describe which layer of the model the river impinges upon.

A form is presented (Figure B.17) for each reach in the vector network, here the user gives values for reach depth, vertical hydraulic conductivity, river width and river bed thickness. These correspond to the values need to calculate head difference between the reach and the element head upon which it impinges as described in Section 5.4.1.


```

Enter a valid site file for well data:

enter a valid site file name
Tern1_group-----

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.12: Well site file selection screen.

```

Enter whether to use this site:

well description is: Hopton
well position is 359320 east and 326620 north
this is row 95 column 5

include well in model, y/n? y

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.13: Well check screen.

```

Enter flows for well for each stress period:

site is: Hopton

      stress      flows
      period

      1           0_-----
      2           0_-----
      3           0_-----
      4          6000_-----
      5          6000_-----
      6           0_-----
      7          6000_-----
      8           0_-----
      9           0_-----
     10           0_-----

next period to edit: 11_-----

      AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

```

Figure B.14: Well flow specification screen.

```

Enter fraction of flow from each level

site is: Hopton
flow fractions should total no more than 1!

      layer      contribution
      number      fraction

      1           0_-----
      2          0.25_-----
      3          0.25_-----
      4           0_-----
      5          0.25_-----
      6          0.25_-----

next layer to edit: 7_-----

      AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

```

Figure B.15: Well layer specification screen.

```

Enter the main river parameters:

      give a vector file representing river network
      river_tern_____

single
river
height
      y      0.0000000000      _____ y

single
river
layer
      y      1_____      _____ y

      AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

```

Figure B.16: River vector selection screen.

```

Enter the parameters for reach: 0      (out of 13 reaches )
Label: 1 Category:
River Tern

Give a depth for this reach
3_____

Give vertical conductivity for the bed of this reach
1500_____

Give width of this reach
10_____

Give bed thickness of reach
1_____

      AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

```

Figure B.17: River reach specification screen.


```

enter the mode of
evapotranspiration (1 or 2) :1

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.18: Evapotranspiration check screen.

B.1.5 Evapotranspiration option

In the first form (Figure B.18) the user specifies the mode of evapotranspiration, either 1 or 2 (McDonald and Harbaugh, 1988).

The second form (Figure B.19) is presented to the user for each stress period within the model. Here the user specifies the height of maximum evapotranspiration, the maximum ET rate and the extinction depth — the depth below which no more evapotranspiration takes place.

B.1.6 Recharge option

This is very similar to the ET option, the first screen (Figure B.20) allows the mode of recharge to be specified (McDonald and Harbaugh, 1988).

The second screen (Figure B.21) allows the user to specify a recharge rate for each stress period. If there are more than ten periods the form spans more than one screen.

B.1.7 Successive implicit procedure option

The *successive implicit procedure* (SIP) module is a very simple option, the user is presented with a form (Figure B.22) which allows various iteration parameters to be set.

B.1.8 Parameters option

This option allows the model parameters to be changed. The first form (Figure B.23) is a warning that tells the user that changing the model parameters will affect any data already compiled. The form asks the user if the change should continue and also if after the change old data should be used in the data options.

If the user decides to change the parameters, the original model parameter form is redisplayed (Figure B.24). When this form is submitted the status of the data already entered for each option will change (Figure B.5) from *complete* to *out of date*

```

Enter evapotranspiration details:

period number: 1

maximum ET surf (normally ground surface)
y      0.0000000000 ----- y

maximum ET rate
y      0.0000000000 ----- y

extinction depth (normally 2-3m)
y      0.0000000000 ----- y


next period to edit   2__

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.19: Evapotranspiration specification screen.

```

Enter the mode of recharge:

mode of recharge (1,2 or 3) :      2

1- recharge to the top layer
2- vertical ,distirbution is specified in an array
3- Recharge is applied to the highest active cell in
each column, it will be intercept by a constant head
element.

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Figure B.20: Recharge code check screen.

period no	one value for rain y/n?	if 'y' enter value:	if 'n' enter raster name for values
1	y	0.0000000000	----- y
2	y	0.0000000000	----- y
3	y	0.0000000000	----- y
4	y	0.0000000000	----- y
5	y	0.0000000000	----- y
6	y	0.0000000000	----- y
7	y	0.0000000000	----- y
8	y	0.0000000000	----- y
9	y	0.0000000000	----- y
10	y	0.0000000000	----- y

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

Figure B.21: Recharge specification screen.

maximum number of iterations	50_
number of iteration parameters to be used	5_
acceleration parameter	1_
criterion for convergence	0.01__

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

Figure B.22: Iteration specification screen.

B.2 Other Modflow-GRASS link programs

B.2.1 *mod.cre*

mod.cre non-interactively creates a new empty modflow dataset. The syntax is:

```
mod.cre [-gr] [dataset=name] [north=value] [south=value] [east=value] /  
[west=value] [ns=value] [ew=value] [layers=value] [stress=value] [type=value]
```

Flags:

- g — use an irregular grid file;
- r — use current region to set model parameters.

Parameters:

- dataset — name of new dataset;
- north — north edge of model region, default: current region north edge;
- south — south edge of model region, default: current region south edge;
- east — east edge of model region, default: current region east edge;
- west — west edge of model region, default: current region west edge;
- ns — north/south resolution, default current region north-south resolution;
- ew — east/west resolution, default current region east-west;
- layers — number of layers, default: 1;
- stress — number of stress periods, default: 1;
- type — time-unit type: 1-sec, 2-min, 3-hour, 4-day, 5-undefine, default: 4.

B.2.2 *mod.write*

mod.write is the program which converts a dataset into Modflow input files, the syntax is:

```
mod.write [-wrepoaq] dataset=name
```

Flags:

- w — include well module;
- r — include river module;
- e — include evapotranspiration module;
- p — include precipitation module;
- o — write output only at end of stress period;
- a — use area weight resampling of irregular grid (default is nearest neighbour);
- q — not verbose.

Parameter:

- dataset — dataset to write to Modflow input files.

B.2.3 *mod.run*

mod.run is a script written to simplify running Modflow. The syntax is:

```
mod.run directory
```

directory is the name of the directory in which output should be written, the default is the directory from which the script is run. Output can be very large, sometimes tens of megabytes.

B.2.4 *mod.xtrct*

mod.xtrct writes data from the Modflow output files into the GRASS database. This program will produce several files describing hydraulic heads and drawdowns in different layers, stress periods and timesteps. Files are normally of the form:

```
name.type.layer.period.timestep
```

or, if the *-j* flag set:

```
name.type.layer.julian_day
```

e.g.

```
well1.draw.1.3.3 or well.draw.1.56.
```

The syntax for *mod.xtrct* is:

```
mod.xtrct [-apwsHzjq] output=name [directory=name] \  
[selection=type,layer,period,timestep[,type,layer,period,timestep,...]] \  
[start=value]
```

Flags:

- a — extract all heads and/or draw downs;
- p — prompt before each extraction;
- w — use area weighted resampling (if irregular grid used) default is nearest neighbour;
- s — write output to site file;
- H — insert a hash into site file descriptor;
- z — miss out zero value sites;
- j — use *julian day* output name form;
- q — quite operation.

Parameters:

output — raster family name;

directory — full directory path of printout (/dir/dir/.../dir), default: '.';

selection — which output to produce, give type,layer,period,timestep;

start — start day (used with *-j* flag), default: 0.

B.2.5 *mod.del*, *mod.rename* and *mod.copy*

mod.del, *mod.rename* and *mod.copy* are house-keeping functions, that allow datasets to be deleted, renamed or copied. The syntax for *mod.del* is:

```
mod.del dataset=name
```

Parameters:

dataset — Modflow dataset name.

The syntax for *mod.rename* is:

```
mod.rename input=name output=name
```

Parameters:

input — old modflow dataset name;

output — new modflow dataset name.

Syntax for *mod.copy* is:

```
mod.copy old=name new=name
```

Parameters:

old — old dataset name;

new — new dataset name.

B.2.6 *mod.region*

This command changes the current GRASS region to match the region of the named Modflow dataset. The syntax is:

```
mod.region [-vpc] dataset=name
```

Flags:

-v — suppress verbose execution;

-p — print Modflow region;

-c — change current region to Modflow region.

Parameter:

dataset — Modflow dataset.

B.2.7 *mod.d.grid*

mod.d.grid is a program that draws the irregular grid of a dataset on to the current monitor. The syntax is:

```
mod.d.grid dataset=name [color=name]
```

Parameters:

dataset — dataset name;

color — color for the grid, options: red, orange, yellow, green, blue, indigo, white, black, brown, magenta, aqua, gray, grey, default: gray.

B.2.8 *mod.rem*

mod.rem performs a bulk remove of raster that match a string pattern. The syntax is:

```
mod.rem search_string [doit]
```

If the option “doit” is not include *mod.rem* will only list the files and not remove them.

B.2.9 *mod.ts*

mod.ts extracts rasters created created with *mod.edit* with *-j* flag and writes to stdout a time series of values. The syntax is:

```
mod.ts -udm [family=... ] [coords=... ] [layer=... ] [add=... ]
```

Flags:

- u — print usage of command;
- d — extract drawdowns rather than heads;
- m — print match string for debugging.

Parameters:

- family — raster family name;
- coords — one of more pairs of coordinates of positions to sample;
- layer — layer from which to sample, default 1;
- add — value to add or subtract from time, i.e. to align the time series with other time series., default 0.

Appendix C

SWMS_2D - GRASS link documentation

C.1 *r.swms*

Unlike the Modflow link set of programs there is just one program forming the basic link between SWMS_2D and GRASS. The program, called *r.swms*, uses the GRASS parser to handle user options. The syntax of the command is:

```
r.swms [-qwsedo] [dataset=name] [editor=name] [scenario=value] \  
[x=value] [y=value] [maxit=value] [lunit=name] [tunit=name] \  
[tolth=value] [tolh=value] [ha=value] [hb=value] [dt=value] \  
[dtmin=value] [dtmax=value] [dmul=value] [dmul2=value] [gdist=value] \  
[po=name] [p2h=name] [p2l=name] [p3=name] [r2h=name] [r2l=name] \  
[aqh=name] [bqh=name] [hcrits=name] [root=name] \  
[tprint=value[,value,...]] \  
[sparam=no,thr,ths,tha,thm,alp,n,Ks,Kk,Thk,poptm,width \  
[,no,thr,ths,tha,thm,alp,n,Ks,Kk,Thk,poptm,width,...]] \  
[aparam=no,tAtm,Prec,cPrec,rSoil,rRoot,hCritA,rt,ht,crt,cht \  
[,no,tAtm,Prec,cPrec,rSoil,rRoot,hCritA,rt,ht,crt,cht,...]]
```

Flags:

- q — quiet operation;
- w — write to SWMS_2D input files;
- s — do not save to dataset;
- e — edit dataset;
- d — use defaults where available;
- o — replace sparam/aparam options instead of adding to them.

Parameters:

- dataset — dataset, default: "noname";
- editor — editor, default: "textedit";

scenario — scenario:

- 1 simple 1D grid, from saturated, with wt input;
- 2 simple 1D grid, from saturated, wt and rain inputs;

x — x co-ordinate of point at which model is applied;

y — y co-ordinate of point at which model is applied;

maxit — maximum number of iterations within a time-step, options: 0-200;

lunit — length units, options: cm,m,mm,km;

tunit — time units, options: secs,mins,hrs,days,yrs;

tolth — max absolute water content change between two iterations, options: 0-1;

tolh — max absolute pressure change between two iterations, options: 0-1000;

ha — upper limit of h for hydraulic prop's table, recommended 0.0001 cm, options: 0-100;

hb — lower limit of h for table hydraulic prop's table, recommended 100000 cm, options: 0-100000;

dt — initial time increment, options: 0.00000001-99999999;

dtmin — minimum time increment, options: 0.00000001-99999999;

dtmax — maximum time increment, options: 0.00000001-99999999;

dmul — time-step multiplier, should be ≥ 1 and < 1.3 , options: 1.0-3.0;

dmul2 — time-step multiplier should be ≤ 1 and > 0 , options: 0.0000001-1.0;

gdist — max distance between grid node, options: 0.00000001-10000000000;

po — value of h (h_1 in Feddes et al.'s (1978) model of the root sink term see Section 6.4.2) above which roots extract water from soil, increasing linearly in;

p2h — upper value of (h_{3u} in Feddes et al.'s (1978) model) above extraction is maximum;

p2l — lower value of (h_{3l} in Feddes et al.'s (1978) model) above which extraction is maximum;

p3 — value of the h (h_4 in Feddes et al.'s (1978) model) below which root uptake ends, wilting point;

r2h — potential transpiration rate (set at $0.5 \text{ cm} \cdot \text{day}^{-1}$);

r2l — potential transpiration rate (set at $0.1 \text{ cm} \cdot \text{day}^{-1}$)

aqh — value of aqh in q(GWL) relationship;

bqh — value of bqh in q(GWL) relationship;

hcrits — max pressure head at soil surface;

root — rooting depth across the catchment default;

tprint — times at which to print any values;

sparam — parameters for each soil in each layer:

- no — layer number;
- thr — residual soil content;
- ths — saturated soil content;
- tha — parameter in soil water retention function;
- thm — parameter in soil water retention function;
- alp — coefficient in soil water retention function;
- n — exponent in soil water retention function;
- Ks — saturated hydraulic conductivity;
- Kk — hydraulic conductivity at thk;
- Thk — moisture content at Kk;
- poptm — pressure head below which roots extract water;
- width — of soil layer (extends to watertable if used);

aparam — environmental boundary conditions at a set time:

- no — record number;
- tAtm — time of record;
- Prec — precipitation;
- cPrec — solute conc of rainfall (not used);
- rSoil — potential evapotranspiration rate;
- rRoot — potential transpiration rate;
- hCritA — minimum pressure head at the surface;
- rt — drainage flux across bottom boundary;
- ht — groundwater depth;
- crt — drainage flux;
- cht — concentration;

C.1.1 The edit mode of *r.swms*

If the flag *-e* is set then, after dealing with all the command line options, *r.swms* enters an interactive mode. The main user interface is a simple menu Figure C.1. The user selects and option by typing the corresponding number.

The options 1, 2 and 3 allow the user to view and edit the basic parameter file, the soil parameter file and the atmosphere file. Options 4, 5, 6 and 7 allow the user to view the error file and the input files generated when options 8 or 9 are selected. Option 8 allows the user to check the validity of the files and generate an error file, option 9 checks the validity, writes an error files and writes the SWMS_2D input files. Option 10 runs SWMS_2D. Option 11 allows any UNIX command, script or program to be performed — several simple scripts were written to convert the output into *Gnuplot*¹ input files. Finally option 12 calls an interactive session of the plotting program in which the results can be plotted. Figure C.2 shows a screen shot of a typical session.

¹ Gnuplot is a freely available plotting programming (Liaw and Crawford, 1995)

```

0-finish

which file to edit
1-basic file
2-sparam file
3-aparam file

which files to view
4-errors file -read_only &
5-selector file -read_only -Wp 0 0 -Ws 780 400 &
6-grid file -read_only -Wp 0 0 -Ws 900 300 &
7-atmosphere file -read_only -Wp 0 0 -Ws 800 300 &

command to execute
8-check validity
9-write input
10-run swms
11-run other command
12-go into gnuplot

>

```

Figure C.1: The *r.swms* edit menu.

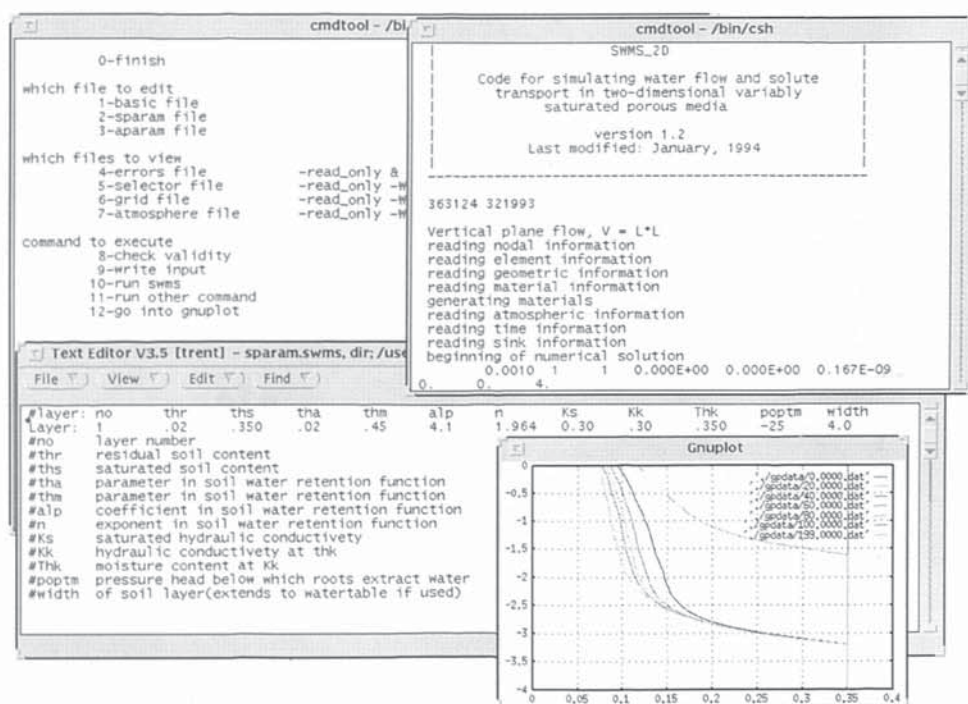


Figure C.2: Screen shot of the *r.swms* edit interface. The top left window is the main edit menu, the top right window has been opened when Modflow was run, the bottom left window is the soil parameter file which has been opened to edit and the bottom right window has been generated by *Gnuplot* to draw the results of the model.

C.2 *r.swms.region*

The program *r.swms.region* was written to apply the program *r.swms* described above (Appendix C.1) across the region of interest. It was written in PERL using the *grass.pl* library (Appendix A.4.1) and the code is documented in Appendix 6.5.2. *r.swms.region* works on a pixel by pixel basis, and can use several machines to perform the individual model runs, the code is documented in Appendix 6.5.2. The syntax of the command is:

```
r.swms.region -uq [machines=... ] [template=... ] dataset=... [error=... ] [sleep=... ]
```

Flags:

- u — print the program usage;
- q — quiet, suppress output.

Parameters:

- machines* — list of machines on which to perform the modelling, default is the local machine;
- template* — a raster that acts to mask out areas not to be modelled, models only in pixels which are not 0 or null, default is no template;
- dataset* — is the swms dataset generated with *r.swms*, default is “noname”;
- error* — name of a file in which to log SWMS_2D error messages, default is none;
- sleep* — period to pause between starting a process and checking it has finished, default is 10 seconds.